PLUS 4 · AMIGA · PC · C64 · C128 · C16 · PLUS 4 · AMIGA

SEPTEMBER 1989 £1.30

YOUR COMMODORE

BATMAN
Waiting for a dark knight

GAMES
- Thunderbirds
- Rockstar ate my Hamster
- Escape from Drill
- Phobia

GREENPEACE
Fight for the Foliage



LICENCE TO KILL
007 – On his own and out for revenge

ISSN 0269-8277
09
9 770269 827007

THE ULTIMATE BACKUP CARTRIDGE GOES ONE BETTER...

NOW ACTION REPLAY Mk V PROFESSIONAL

FOR CBM64/128 HAS ARRIVED



EVEN MORE POWERFUL, MORE FRIENDLY & NOW EVEN MORE FEATURES!

ALL FEATURES AVAILABLE TO TAPE OR DISK USERS.

ACTION REPLAY Mk V differs from all other cartridges because it combines an 8K RAM with a FULL 32K operating system ROM. That means that ALL features are INSTANTLY AVAILABLE AT ALL TIMES.

WARNING!! Other systems use outdated technology which severly limits performance. Action Replay's state of the art hardware gives you MORE POWER, MORE SPEED, MORE FACILITIES than any other cartridge. There really is no comparison. Here are just some of the features...

✓ **TURBO RELOAD** Action Replay Mk V has 2 unique Turbo Loaders - "Ramloader" & "Warp 25". Both work at up to 25 times normal speed! ✓ **PRINTER DUMP** freeze any game & print out the screen. ✓ **PICTURE SAVE** save any HiRes multi colour screen to disk at the push of a button. ✓ **SPRITE CONTROL** full range of controls with 64K operation. ✓ **POKEFINDER GENERAL** an automatic infinite lives generator! ✓ **SUPER COMPACTOR** ultra efficient program compaction techniques. ✓ **TEXT SCREEN EDITOR** modify the text screen on a frozen program. ✓ **SUPERFAST DISK OPERATION** load 200 blocks in just SIX SECONDS. ✓ **MORE TAPE FACILITIES** you don't need a disk drive to use Action Replay. ✓ **TOOLKIT COMMANDS** full range all at the press of a button. ✓ **FULLY INTEGRATED OPERATION** with onboard custom LSI LOGIC PROCESSING CHIP. ✓ **CENTRONICS INTERFACE** for parallel printers. ✓ **PROFESSIONAL MACHINE CODE MONITOR** Full 64K monitor available at all times with full range of commands.

TOTAL BACKUP POWER

The most powerfull backup system ever devised. Unstoppable freezer system. Freeze at any point. Just press the magic button to backup your programs.
❑ TAPE TO TAPE ❑ TAPE TO DISK ❑
❑ DISK TO DISK ❑ DISK TO TAPE ❑
VERY fast & efficient program compaction. Single part save at TURBO speed to disk or tape. Backups turboload INDEPENDENTLY of cartridge. No "programing" or user knowledge required.
THE PROCESS IS AUTOMATIC - JUST GIVE THE BACKUP A NAME.

WHAT THE REVIEWERS SAID

"I'm stunned, amazed and totally impressed. This is easily the best value for money cartridge. *The Cartridge King!*"
Commodore Disk User

UPGRADE INFORMATION

Mk IV Professional to Mk V Professional - just send £9.99 & we will send you a new Mk V Operating System Chip for you to just plug in! Mk IV (Standard) to Mk V Professional - send your old cartridge plus £15.99 & we will upgrade it (allow 14 days).

PERFORMANCE PROMISE

Action Replay has an unmatched range of onboard features. When you buy Action Replay, if you don't find our claims to be true, then return it within 14 days for a full refund.

GRAPHICS SUPPORT UTILITIES DISK

SLIDE SHOW. View your favourite screens in a slide show type display.

BLOW UP. A unique utility to allow you to take any part of of your picture & 'blow it up 'to full screen size.

SPRITE EDITOR. A complete sprite editor helps you to create or edit sprites.

MESSAGE MAKER. Takes your favourite screen created with a graphics package or captured with Action Replay & turns it into a scrolling screen message complete with music.

ONLY £12.99

NOTICE 1988 COPYRIGHT ACT

DATEL ELECTRONICS Ltd. neither authorizes or condones the use of it's products to reproduce copyright material. It is illegal to make copies of such material without the expressed consent of the copyright owners or thier licencees.

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

CALLERS WELCOME - Please reserve goods by telephone prior to visit.

DATEL ELECTRONICS

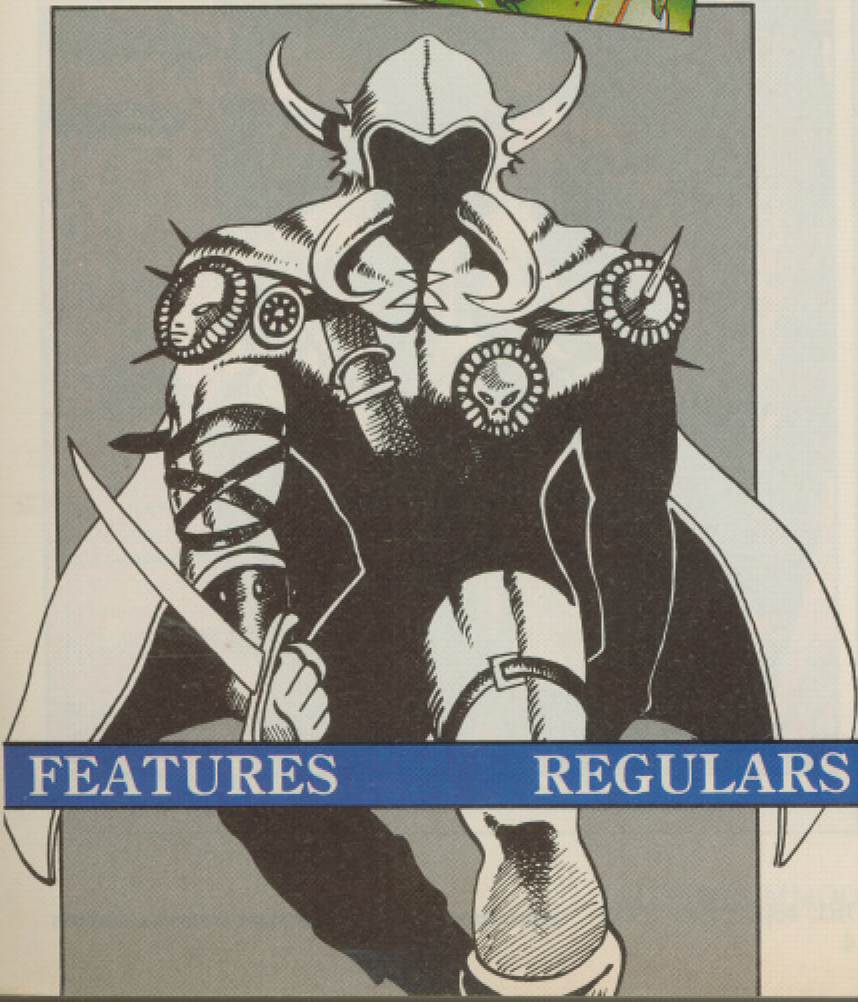
DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324









CONTENTS

FEATURES REGULARS PROGRAMS GAMES

Editorial

If you flick through the pages of this month's *Your Commodore* you may find a few subtle (and a couple of blatantly obvious) changes. This is because the editorial team, upon my arrival to the fold, has decided to offer a brighter future for *YC,* one that we can all enjoy together.

My first major decision was to revolve this issue around Batman, and I hope you all agree that this was a wise move. Apparently the movie grossed over $40,000,000 in its first weekend, and seeing as it cost $10m less to make it has more than dispelled any fears from Warner Brothers that it was to flop, and this is before it reaches British shores. Further on in this issue you can read about the reasons the film was undertaken, and a preview of the game.

Other features of note are on Licence to Kill – the new Bond film, Greenpeace and Lone Wolf, so make sure you've got your asbestos gloves on, they are HOT!

Although there are many entertainment based articles we haven't left out the usual listings, or the machine based columns. The programs can now be found in a section of with their own, so you can keep them separate from the rest.

I hope you enjoy this issue as much as we've enjoyed putting it together and that leaves me to say, welcome to the beginning...

Rik Henderson

Deputy Editor: *Richard 'Boy Wonder' Henderson*
Technical Editor: *Paul 'Riddler' Eves*
Group Editor: *Stuart 'Penguin' Cooke*
Advertisement Manager: *Paul 'Batpaole' Kavanagh*
Sales Executive: *Maria 'Catwoman' Wade*
Ad-Copy Control: *Karen 'Batgirl' Bashford*
Artists: *Alan 'Batdog' Batchelor, Alan 'Joker' Lathwell*
Designer: *Mark 'Dark Knight' Newton*
Origination: *Ebony*
Printers: *Chase Web*

Distribution SM Distribution, 6 Leighham Court Road, London SW16 2PG. Printed by Chase Web Plymouth.

Your Commodore incorporating Your 64 is a monthly magazine appearing on the first Friday of each month. Argus Specialist Publications Limited. Editorial and Advertisement office, *Your Commodore,* Argus House, Boundary Way, Hemel Hempstead HP2 7ST. Telephone: (0442) 66551. Subscription rates upon application to Your Commodore Subscriptions Department, Infonet Ltd., 5 River Park Estate, Berkhamsted, Herts HP4 1HL. Telephone: (0442) 872279. U.S.A. Subscription Agent: Wise Owl Worldwide Publications, 4314 West 238th Street, Torrance CA 90505 U.S.A.

Data Statements

The Bug Bites Back

Now that The Bug, the incredibly successful computer fanzine, has officially folded, editors Jaron Lewis and Jeffrey Davey are to sell their entire software review library to raise funds for charity.

The collection includes five years of software (leisure and otherwise) and a majority of the profits are to go to a consortium of separate charities and voluntary organisations.

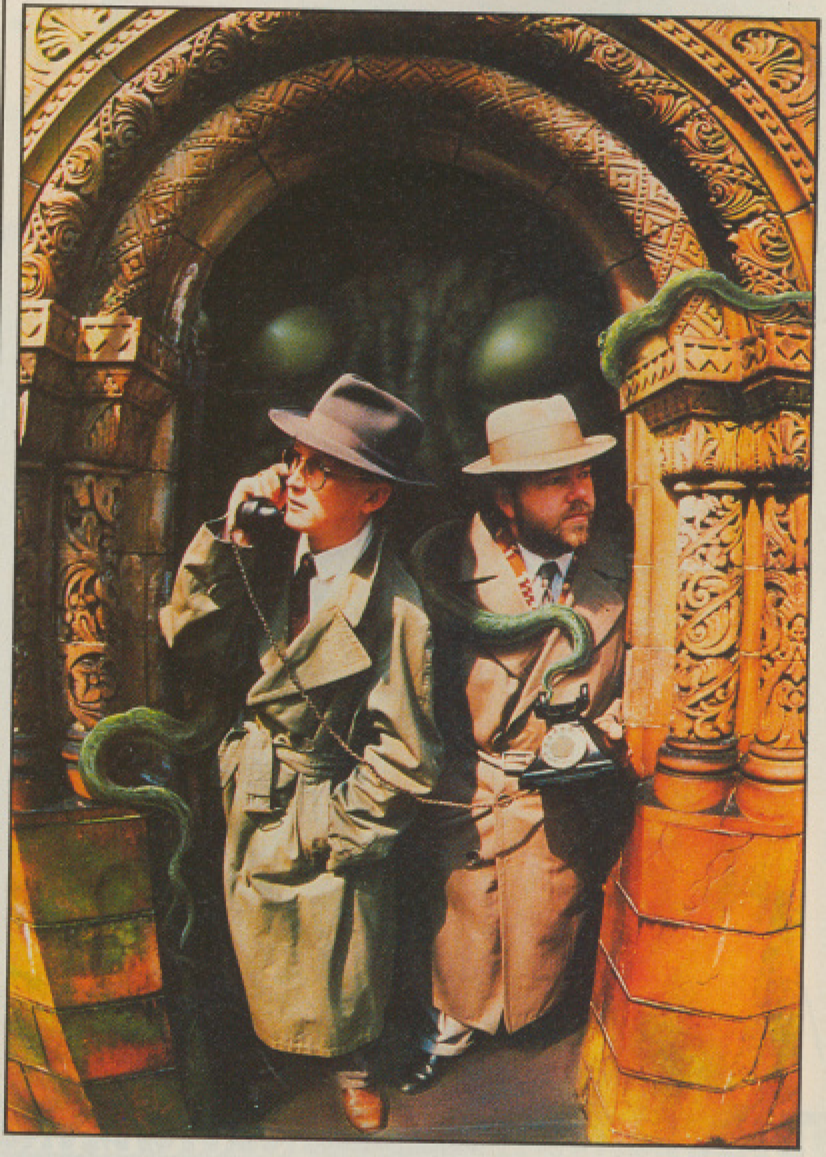
If you wish for more information, the current list, etc. write to 28 Leaside Avenue, Muswell Hill, London N10 3BU. Please include an SAE.

EA's Horror Story.

Electronic Arts is to bring the horrific world of H P Lovecraft's Cthulhu mythos onto Amiga and PC screens in September. The Hound of the Shadow is a role playing game set in the roaring 20s, in which the player must use investigative research in order to unravel the secrets of arcane lore and discover many odious horrors lurking beneath seemingly normal life.

The game system that was developed by board game designers and fiction writers, Eldritch Games, allows the player to create a character by selecting sex, nationality, one of six professions, and proficiency in over 50 skills. These will directly affect the outcome and his success in the game.

The Hound of the Shadow will appear first on the PC in early September and later on the Amiga. Both versions will cost £24.99



OFFICIAL COMMODORE/AMIGA DEALER

COMPUTERS	
Amiga B2000 only	699.00
Amiga B2000 + 1084S Stereo Col Mon	899.00
Amiga 2088 PC/XT Bridgeboard + 5.25" Drive	299.00
Amiga 500 + Modulator	359.00
Amiga 500 + Super Software Pack	389.00
Amiga 500 + 1084S Stereo Col Mon	609.00
Amiga 500 + Philips 8833 Stereo Col Mon	589.00
Commodore PC1 Single Drive Mono	332.35
Commodore PC10/20/40/60	PHONE
Commodore 128D Desktop	379.00
Commodore 64C Brainbox Pack NEW (64C, Cassette, J/stick, 10 Games)	139.99
B2000 + 1084S + XT B'Brd + 20 MBHD	1329.80
PRINTERS	
Citizen 120D Parallel or Commodore	129.95
Star LC-10 Parallel	189.00
Star LC-10 Commodore inc lead	215.00
Star LC-10 Colour Par or Comm	245.00
Star LC24-10 24 pin Multi-font 170/57cps	329.00
All Okimate 20 consumables normally in stock	PHONE
MONITORS	
Commodore 1084S Colour Monitor inc lead	259.00
Philips 8833 Stereo Colour Monitor inc lead	229.00
Philips 7502 Green Screen Monitor	75.00
MISCELLANEOUS	
Amiga M501 Microbotics 512K RAM/Clock Exp	119.95
Amiga 500 AMDRIVE 30Mb Hard Disk Drive	399.00
Amiga 590 20Mb Autoboot Disk Drive. 2Mb RAM Expandable	399.00
miniGEN Genlock for Amiga	113.85
Pace Modem + S/ware for Amiga V21/23 Prestel, BBS etc.	239.00
Commodore 1541-II Disk Drive	139.00
Commodore C2N Data Recorder	29.95
Cumana CAX354 Amiga Disk Drive	£95.00
Power Supply for C64	26.45
Super-G Cent I/F + Graphics C64/128	34.95
Surge Protector 13A Plug	12.95
Surge Protector 3-Way Adaptor/4-Way Dist Unit	19.95/15.95
C128 Official Prog's Ref Guide	p&p £2 24.95
SOFTWARE	
VizaWrite 128 + Spellcheck	49.95
VizaStar 128 Spreadsheet & Database	69.95
Superbase 64/128	24.95/29.95
Superscript 64/128	24.95/29.95
Tasword 64 40/80 Col WP - Tape/Disk	24.95/29.95

★ **ALL PRICES ARE INCLUSIVE OF VAT AT 15%**
★ **CARRIAGE £5 (EXPRESS £10). Software and small items free**
★ Prices subject to change without notice. E&OE

Delta Pi Software Ltd

8 Ruswarp Lane, WHITBY, N. Yorks. YO2 1ND.
Tel: 0947 600065 (9am-7pm)



WIN WITH COURSEMASTER

NEW

THE COMPUTER HORSERACING PROGRAMME

- ★ RATES ANY RACE IN SECONDS - ANY DAILY NEWSPAPER IS ALL YOU NEED
- ★ NEVER out of date - Both N. Hunt and Flat - Fast data entry
- ★ AMAZING ACCURACY!! - Now you **CAN** BEAT THE BOOKIES!!
- ★ Works on the simple princilpe that **FAST HORSES BEAT SLOW ONES!!**
- ★ Clearly identifies best selection in every race plus these **AMAZING** Features:-
- ★ First, Second and Thrid choice shown for forecasts and Tricasts etc. Recommends most suitable type of bet.
- ★ Actually works out your WINNINGS on most popular bets including SINGLES and DOUBLES, win and each way, PATENTS, YANKEES, CANADIANS, HEINZ etc. Good EACH WAY and LONG ODDS bets clearly shown
- ★ Will PRINT out your BETTING SLIP for you
- ★ Maintains a BANK ACCOUNT - BET Like PROFESSIONALS Do! Records all your bets in any number of accounts. Keep a complete record of your betting or compare COURSEMASTER against your favourite tipster.
- ★ PLUS!! - THE **AMAZING COURSEMASTER SYSTEM**. This superb betting system is included in the programme. A system which regularly produces huge wins from small stakes. Try it! Can pay for itself many times over on the first day!
- ★ Supplied witth 20 page BETTING GUIDE and MANUAL

FREE HOT TIP OF THE MONTH SENT TO EVERY PURCHASER

Supplied on TAPE at £14.95 for:- All Spectrums, Commodore 64/128, Amstrad CPC's, BBC & Electron Also Available for Sinclair QL on Microdrive at £15.95.

Supplied on DISC at £16.95 for:- All Spectrums, Commodore 64/128, Amstrad CPC's, Amstrad PCW's, BBC & Electron, IBM/PC compatibles, All Amstrad PC's etc.

Please state machine and disc size

INTRASET LTD. (DEPT YC), FREEPOST (No Stamp needed) 6 Gilderdale Close, Gorse Covert, Birchwood, Warrington, Cheshire, WA3 2BR. Or large SAE for further details.

NEW FOR YOUR COMMODORE FROM TRILOGIC!

TRILOGIC

AMIGA AUDIO DIGITISER

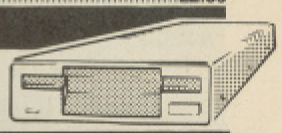


— NEW, CHOOSE MONO OR STEREO VERSION

Both Amiga audio digitisers give superb performance, unsurpassed at the price. No software is supplied, since they are fully compatible with Perfect Sound, Prosound Designer, Audiomaster, and Datel's Prosampler. Sampling rates up to 60KHz are possible depending upon the software. An audio lead is supplied for connecting to the headphone socket or line output of a radio, personal stereo, keyboard etc. Full instructions are included, and the mono version also has an L.E.D. overload indicator. A public domain "Sound Workshop" disk is available which has demo versions of Audiomaster & Perfect Sound etc.

MONO DIGITISER .... £27.99 POST FREE   STEREO DIGITISER £37.99
SOUND WORKSHOP DISK .... £4.99 POST FREE IF PURCHASED WITH DIGITISER
ADAPTOR FOR A1000 .... £2.00

EXTERNAL 3.5" DISK DRIVES

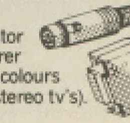


- 880K FORMATTED CAPACITY
- THROUGH POST
- VERY QUIET OPERATION
- SLIMLINE STEEL CASE
- LOW POWER CONSUMPTION

3.5" DRIVE WITH ON/OFF SWITCH .... £79.99 POST FREE

AMIGA MONITOR & PRINTER LEADS

RGB TV & MONITOR LEADS



We have leads to connect all AMIGAS to your TV or colour monitor provided it has an RGB input socket. All leads give a much clearer picture than using the AMIGA MODULATOR, permit ALL 4096 colours to be displayed and include the audio lead (to give stereo with stereo tv's).

ORDER AL 1 FOR TV's WITH 21 PIN EURO (SCART) SOCKET.
FITS PHILIPS, FIDELITY, SONY, GRUNDIG, NORMENEDE, ETC. .... ONLY £9.99
ORDER AL 2 FOR FERGUSON TV's WITH 7 or 8 PIN DIN SOCKET.
MODELS MC01 & MC05, ETC. .... ONLY £9.99
ORDER AL 4 FOR HITACHI & GRANADA TV's WITH 7 PIN DIN SOCKET. MODELS CPT1444, ETC. .... ONLY £9.99
AMIGA TO MULTISYNC MONITOR (analogue RGB) AL11 .... £9.99

MANY OTHER AMIGA MONITOR/TV LEADS LISTED IN OUR FREE CATALOGUE.
OUR LEADS ARE GUARANTEED TO WORK WHERE OTHERS DON'T!
LEADS ALSO AVAILABLE FOR ATARI ST RANGE. PLEASE CONSULT US IF IN DOUBT.

NEW — MONITOR SHARER

SHARE 2 COMPUTERS WITH ONE TV/MONITOR
Plug-in the scart leads from any two computers eg AMIGA & ST, and connect the SHARER to the tv or monitor's scart socket. Just press the switch to instantly select either computer .... MSWI £15.99

OTHER LEADS

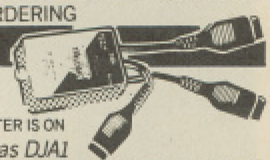
AMIGA 64 EMULATOR LEADS	AEL 1 ONLY £4.99
MODULATOR EXTENSION LEADS	MEL 1 ONLY £8.99
AMIGA PRINTER LEAD — Parallel Type 1.5M long	AMP 1 ONLY £6.99
3.0M long	AMP 3 ONLY £9.99
AMIGA A500/BBC EMULATOR LEAD	BE1 ONLY £7.99
AMIGA A500 TO SERIAL (RS232) PRINTER 1.5m	RS1 ONLY £9.99
AMIGA A500 TO MODEM (25W 'D') 1.5m	RS6 ONLY £10.99
AMIGA NULL MODEM LEAD — LINKS TWO A500s 1.5m	RS5 ONLY £7.99
AMIGA TO HIFI LEAD 3m 2 PHONO PLUGS EACH END	HL1 ONLY £4.99
AMIGA TO HIFI LEAD 5m 2 PHONO PLUGS EACH END	HL2 ONLY £5.99

OTHER LENGTHS OF LEAD AVAILABLE — PLEASE SEE OUR CATALOGUE

EXTERNAL DRIVE SWITCH FOR DF1	ADE3 ONLY £8.99
EXTERNAL DRIVE SWITCH FOR DF1 & DF2	ADE4 ONLY £10.99

PLEASE STATE WHICH AMIGA YOU HAVE WHEN ORDERING

MOUSE/JOYSTICK SWITCHER



- NO MORE UNPLUGGING MOUSE WHEN JOYSTICK REQUIRED
- EXTENDS THE PORT FOR EASIER ACCESS
- SAVES WEAR & TEAR ON THE MOUSE PORT
- PUSHBUTTON SWITCH CAN BE OPERATED WHEN THE COMPUTER IS ON

A VERY USEFUL GADGET — ONLY £10.99 order as DJA1

AMIGA MINI STEREO AMPLIFIERS



MINIAMP 2 combines a mini stereo power amp with two neat speaker units which connect directly to your AMIGA. They are ideal for use with mono TV's & monitors, & simply plug in for instant stereo sound. You'll be amazed at the difference.
MINIAMP 2 WITH REMOTE VOLUME CONTROL & ALL LEADS **ONLY £19.99**

MINIAMP 4 comprises separate 4 watts per channel high quality mini stereo amplifier with loudspeaker/headphone socket, separate speaker units with 4" twin cone drivers, mains power pack & connecting lead for AMIGA. CAN BE USED AS 'WALKMAN' BOOSTER & WITH OTHER COMPUTERS

MINIAMP 4 STEREO AMPLIFIER SPEAKER SYSTEM	MA4	£34.99
MAINS POWER PACK FOR MINIAMP 2	MPP1	£5.99
LIGHTWEIGHT QUALITY STEREO HEADPHONES	SHP1	£5.99

NEW — AMIGA MIDI INTERFACES

MIDI INTERFACE 1 FOR A500/2000 OR A1000 .... **MIF1 £29.99**
This has MIDI IN, OUT & THRU sockets. Self powered.
MIDI INTERFACE 2 FOR A500/2000 OR A1000 .... **MIF2 £34.99**
This has MIDI IN, MIDI OUT, MIDI THRU plus 2 UNIQUE switchable MIDI OUT/THRU SOCKETS to save you having to swap MIDI leads when using more than one MIDI instrument. Self powered.

SUNDRY ITEMS — BARGAIN PRICES

SONY 3.5" DISKS 100% Guaranteed DS/DD disks

Lowest Prices ... Per 10 **£12.99 & Postage**   Lowest Prices .... Per 25 **£28.49 Post Free**
Lowest Prices .... Per 50 **£55.99 Post Free**   Lowest Prices .... Per 100 **£99.99 Post Free**

3½" DISK BOX FOR 50 DISKS. SMOKED LID, LOCKABLE, DSB4	ONLY £8.45
3½" DISK BOX FOR 100 DISKS. SMOKED LID, LOCKABLE, DSB6	ONLY £12.49
A500 DUSTCOVER. ANTISTATIC TYPE, ADC1	ONLY £4.99
A4 NON-SLIP MOUSEPAD, AFM1	ONLY £4.99

WE ALSO STOCK AMIGAS, PCs, PRINTERS, MONITORS, HARD DISKS, MEMORY EXPANSIONS, VIDEO DIGITISERS, SELECTED SOFTWARE & BOOKS — PHONE FOR OUR FREE 20 PAGE CATALOGUE. TRADE ENQUIRIES WELCOME. Please add 75p part postage + packing to orders under £15.00

FREE CATALOGUE WITH ALL ORDERS *EXPRESS DELIVERY £1.99 Extra*

TRILOGIC

TRILOGIC, Dept. Y.C. Unit 1.
253 New Works Road, Bradford BD12 0QP
Tel 0274-691115 Fax 0274 600150

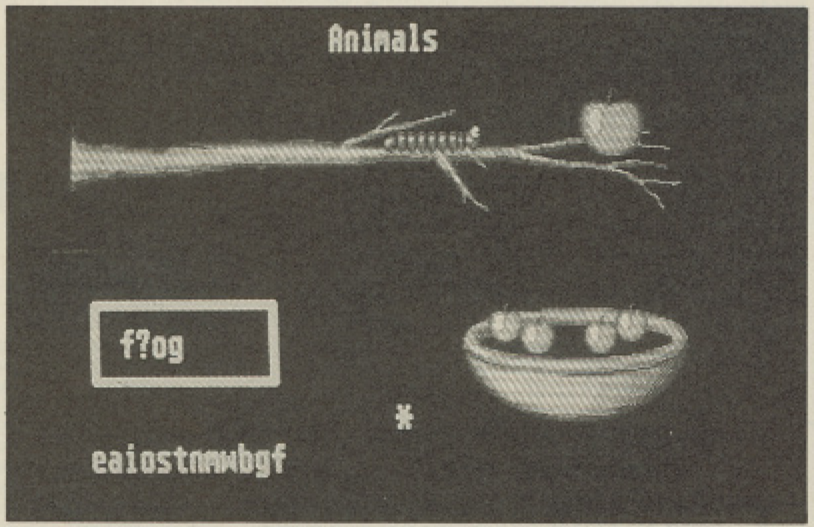
· FAST MAIL ORDER SERVICE · PROMPT DESPATCH · ALL PRICES FULLY INCLUSIVE PLUS 10 DAY MONEY BACK GUARANTEE ON ALL ITEMS. PAYMENT BY CASH, CHEQUES PAYABLE TO TRILOGIC BY POSTAL ORDER, ACCESS OR VISA. ADD £2·99 EXTRA FOR EXPORT ORDERS, PAYMENT IN STERLING ONLY PLEASE.

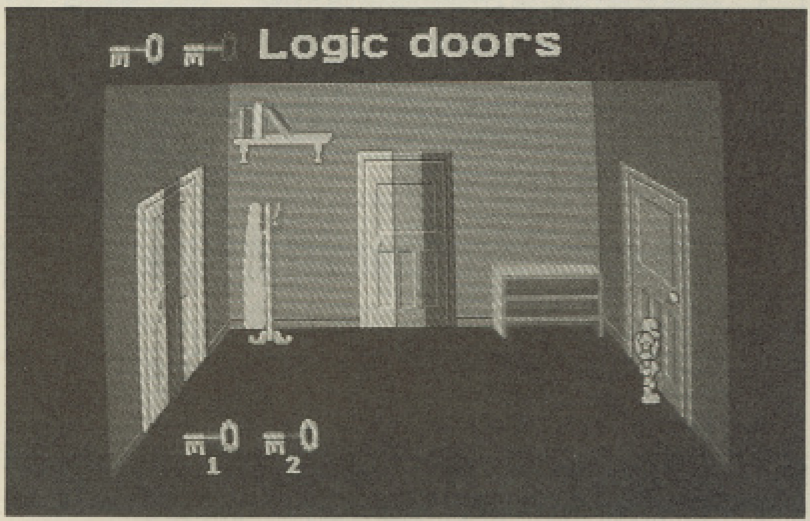
Trendy Lefties in Fun School Controversy

Fun School 2, Database's excellent teaching software, has been rejected for use in schools by 'Trendy' authorities, who believe that it emphasises the 3Rs too much.

The 3Rs (Reading, 'Riting, and 'Rithmatic) are currently being weedled out of the national curriculum by such left-wing educationalists for the 'old fashion' methods of teaching used. Peter Davidson, one of the co-authors of the software, states "What they don't like about it – apart from it being based on the 3Rs – is that it involves an element of competition, with a tick coming up if the question is answered correctly".

Shelly Gibson, a teacher who has been using Fun School 2 in her class at Poplar Street Primary School, Audenshaw, has written to the Minister of Education Kenneth Baker, to urge him to explore matters further.







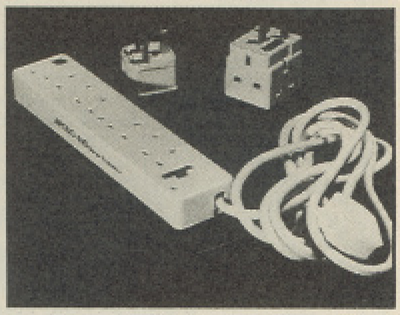
PC Formula One.

Electronic Arts is to release a PC version of its best selling Amiga racing game, Ferrari Formula One. Planned for an early August date, the game aims to bring together the excitement of driving in a Formula One championship against Alain Prost, Nigel Mansell and Ayrton Senna and the strategy of managing a team.

The game will feature all 16 tracks from the 1986 season including Monaco, Brand's Hatch and Monza plus two additional tracks so that the player can compete in the current season. Each track is modelled accurately down to its length, the background and the weather conditions the drivers will face.

The races themselves aim to put you in the middle of the action as you attempt to drive your Ferrari to victory. Ferrari Formula One will cost £29.99. Available on both 3.5 and 5.25 disks it will support both CGA and EGA graphics.

Socket to Them



Computers, monitors and TVs are all vulnerable to mains interference. This can damage hardware, corrupt data, and cause games to crash, not ideal when the high score is about to be broken. The most common type of inteference is a mains 'spike', provided by home equipment such as washing machines, power tools, and microwave ovens. The same applies with your neighbour's electrical equipment, or even the Electricity company having problems maintaining a stable voltage.

One solution to such a problem comes from Apollo Electronic Products Ltd. who have introduced a new Apollo Spike Suppressor which, for the price of £12.50 for a plug unit, can protect your computer system. These British made devices are simple to use, the most you have to do is replace the mains plug.

The range also includes a 3-way adaptor for £16 and a 4-gang strip for £35 (prices incude P&P and VAT). All the units are capable of absorbing a spike of 4500 Amps/225 Joules.

For more information contact Apollo Electronic Products Ltd. on (04858) 8156.

Game, Set and Mat

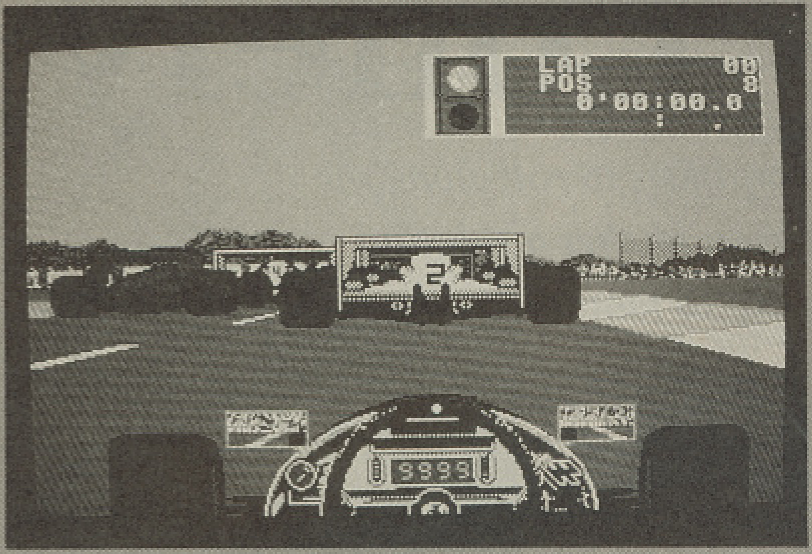
Your Commodore are proud to present mouse mats and dust covers for the A500 and C128. The covers are especially designed to protect your computer from dust, grime, coffee, tea, etc, ensuring your circuits do not get zapped by spillages and internal acid rain.

Made exclusively for your favourite computer mag, they are stylish and robust, and come complete with their own nifty logo.

The mouse mats too are emblazoned with the 'Your Commodore' logo, and they help to allow your mouse grip on any surface.

At the price of £4.45 for the covers and £4.45 for the Mouse Mat, they represent terrific value for money.

They are available from: **Readers Services, Argus House, Boundary Way, Hemel Hempstead HP2 7ST.**

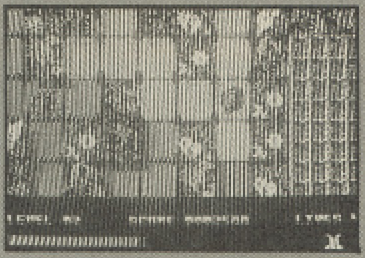


It's A-Maze-ing

Hewson's latest game comes our way in late August. Titled Mazemania it features a character called Flippo who must travel through the maze of Tubular Bells turning tiles as he goes – hence his name Flippo.

Gliding aliens hinder his quest, but there are icons to supply him with extra power. Once gyrated, the maze will fade away to leave Flippo with yet another to tackle.

The Amiga version will have 20 levels to complete, whilst the C64 will only feature 12-16. The price of the C64 version will be (on cassette) £9.99.

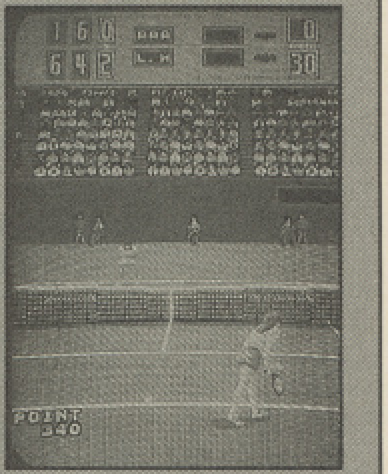


Anyone for Tennis

Mirrorsoft, under their Imageworks label, has secured the rights for the Sega coin-op, Passing Shot. Programmed by Teque – the team behind Thunderbirds and Blasteroids, among others – it captures all the options of the arcade machine and will be available on both the C64 and the Amiga.

Passing Shot features doubles and singles championships, taking place on clay and grass courts, in an effort to win the elusive Grand Slam. It offers an impressive variety of gameplay options with the player controlling both the positioning and type of shot (slice, lob, spin, etc.). The court scrolls to allow you to follow the action and there are two perspectives to encounter during the game – overhead for gameplay, 3D for service.

Ball boys, the net judge and the linemen are all there and we can expect to see it in late Autumn.



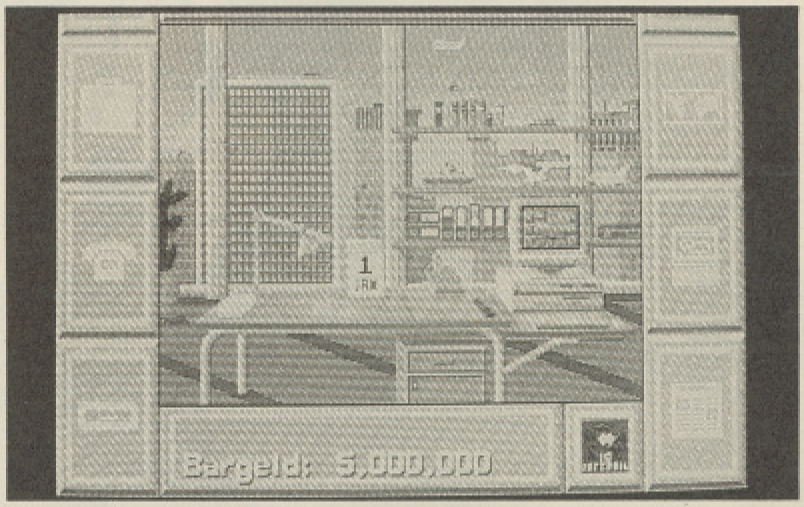
Oil See You in the Morning

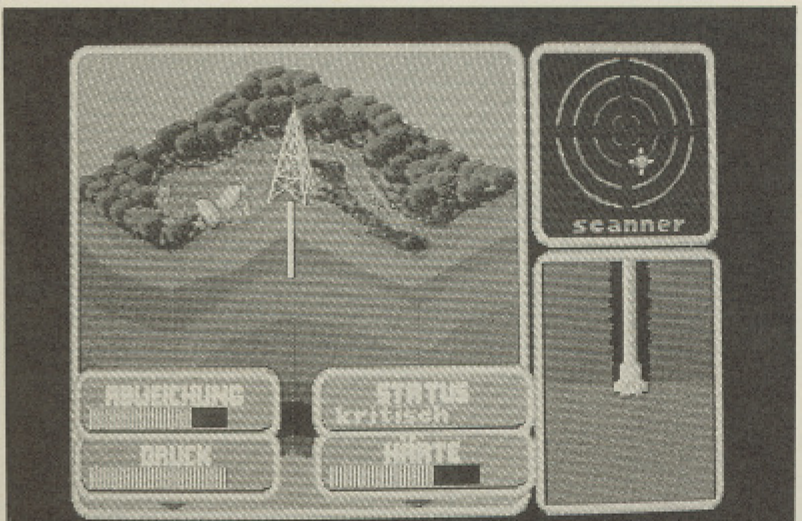
Rainbow Arts, the company responsible for the scandalous Denaris/Katakis, has undertaken a different kind of project.

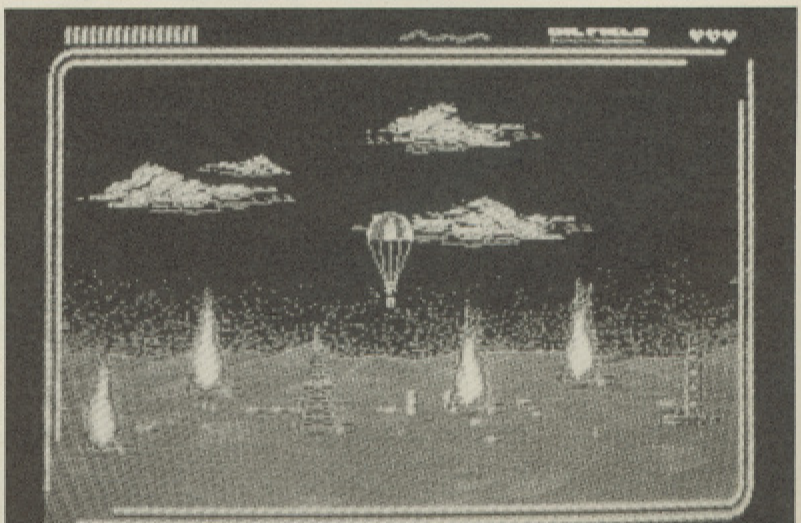
Released on the Reline Software label, Oil Imperium deals with the cut and thrust world of the oil business. You can become J.R. as you try to put your competitors out of business. What is even better is that these enemies can be your friends, or the computer.

Bullying, selling, trading, spying and sabotaging are all involved in getting to the top, so only the devious minds may apply.

Oil Imperium will be available in September on the C64 and Amiga!

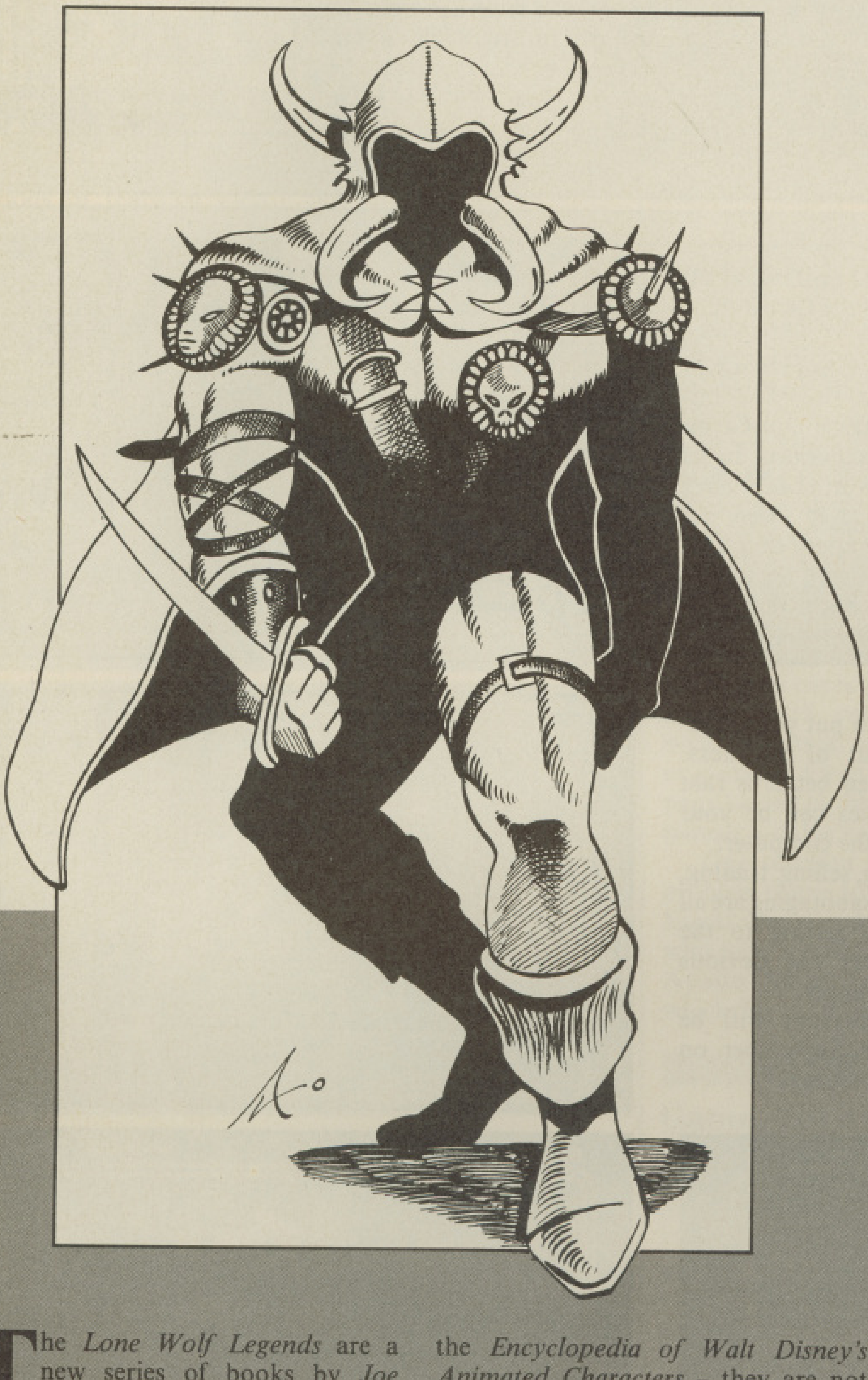






Only the Lonely

The Lone Wolf bandwagon continues to roll along the road of success. Rik Henderson hitches a ride and prepares for a one-way trip to fantasy land.



The *Lone Wolf Legends* are a new series of books by *Joe Dever*, who hopes to emulate the success of previous efforts. Co-written with John Grant – whose credits to date include *Dreamers, Earthdoom* (co-written with David Langford) and, strange as it may seem, the *Encyclopedia of Walt Disney's Animated Characters* – they are not short of technical brilliance, although one wonders whether the fantasy orientated plot is a mite too clichéd.

Eclipse of the Kai is the first in the indeterminable number of books. It deals with the origin of Lone Wolf plus the downfall of the Kai, the warrior elite of Sommerlund (as it quotes on the back). Although it contains many good fantastical ideas, there is always a feeling of deja-vu connected.

The Dark Door Opens is not much better in theory, but perhaps a more enjoyable read. Now that the Kai have been eliminated, Lone Wolf is born and the plot follows his quest to reach King Ulnar and warn him of impending doom.

Audiogenic Software has now sussed the effect of Lone Wolf, and thus released a new computer game based on his exploits.

Entitled simply *The Mirror of Death*, the game places Lone Wolf in an arcade-adventure. Although one could be forgiven for thinking that it has more than a smidgen of hack-and-slash action more usual in a beat-em-up.

The plot follows Lone Wolf as he tries to recover a stolen gem, once belonging to his mentors, the Kai. This is no easy task as the tower in which it is imprisoned is full of fire-spitting gargoyles, demons and countless other fatal dangers. Once all these have been negotiated, Wolf comes face-to-face with the Mirror of Death itself, which is not termed such for nothing. It has many properties, but its most famous, and arguably its most lethal, is the fact that it can produce carbon-copies of its enemies. Not just any old reproduction mind you, but one that has evil intentions. It will take more than a quick slash to vanquish this fellow.

Lone Wolf – The Mirror of Death boasts great graphics and good action. The only reservation that I have is that the play may not be varied enough, thus giving it a short life.

It is available for the PC, Amiga, and C64 at varying prices.

Other news comes our way that *Mirror of Death* will not be the last Lone Wolf excursion into the world of home computers, *Audiogenic* has many more adventures planned for

1990.

Last, but certainly not least, in the new Lone Wolf releases is the inter-



active telephone line, written by Joe Dever. This is surely in direct competition with *Steve Jackson's Fist*.

The company involved, *Broadsystem*, has invented a new label under which they hope to release many games of this kind, *Fortress of Doom* being just the first. *Phonequest* (TM) games will all use the same system of operation, although at this stage it is hard to determine whether they'll just be sticking with Lone Wolf, or if they may branch into other areas of interactive 'quests'.

Unlike *Fist*, which uses a tone method to determine actions, *FOD* breaks new ground with voice-interactive technology. This allows the caller to make 'real-time' decisions during play, and increases the tension created by the stunning sound-effects.

The way in which this works is simple. When at a point in the game where a passive decision is to be made, you will be given a number of options (like the Lone Wolf gamebooks), each separate option carries a different number which must be quoted after a bleep, thus choosing the action to be performed. Aggressive moves are similar, although they must be quoted when in battle, else a sticky end is almost a certainty.

The sound-track was recorded at a top London studio using a very able team of professional actors, with many other effects added to make the game an all-round fantasy experience. Separate sections of the adventure are narrated by different voices (male and female) and they all do a great job in the atmosphere stakes.

One regret I have of any game like this is the prices which must be charged. Off peak rate is 25p per minute, whilst peak is 38p per minute, costing a mammoth £15 for a hour's play (off peak), which will add up to £105 per week, if it is played for an hour a day, quite a sum of money.

This is most worrying for the younger children who become addicted to such things (hence the reason for *Chatline's* demise). My advice to anybody worried about the costs, buy one of those gamebooks, at around £2.50, and get a friend around to read it out to you, it's not that much different and certainly a lot cheaper.

I'm not here to condemn the line, I think it's rather good fun, and if played in moderation it's a very worthwhile venture.

All the Lone Wolf products have something going for them, and the fact is that they'll be popular for a long while yet, thanks to *Joe Dever's* persistence, and *Beaver Books* money, for when both are flowing legends can be born.

Lone Wolf Bibliography

Lone Wolf Gamebooks

1. Flight From The Dark – *Joe Dever & Gary Chalk.*
2. Fire on the Water – *Joe Dever & Gary Chalk.*
3. The Caverns of Kalte – *Joe Dever & Gary Chalk.*
4. the Chasm of Doom – *Joe Dever & Gary Chalk*
5. Shadow on the Sand – *Joe Dever & Gary Chalk*
6. The Kingdoms of Terror – *Joe Dever & Gary Chalk*
7. Castle Death – *Joe Dever & Gary Chalk*
8. The Jungle of Horrors – *Joe Dever & Gary Chalk*
9. The Cauldron of Fear – *Joe Dever*
10. The Dungeons of Torgar – *Joe Dever*
11. The Prisoners of Time – *Joe Dever*
12. The Masters of Darkness – *Joe Dever*

The World of Lone Wolf Gamebooks

1. Greystar the Wizard – *Joe Dever & Ian Page*
2. The Forbidden City – *Joe Dever & Ian Page*
3. Beyond the Nightmare Gate – *Joe Dever & Ian Page*
4. War of the Wizards – *Joe Dever & Ian Page*

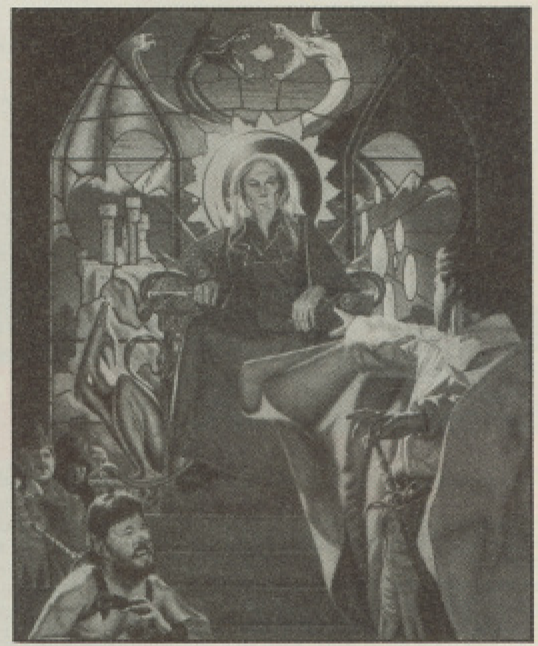
Legends of Lone Wolf

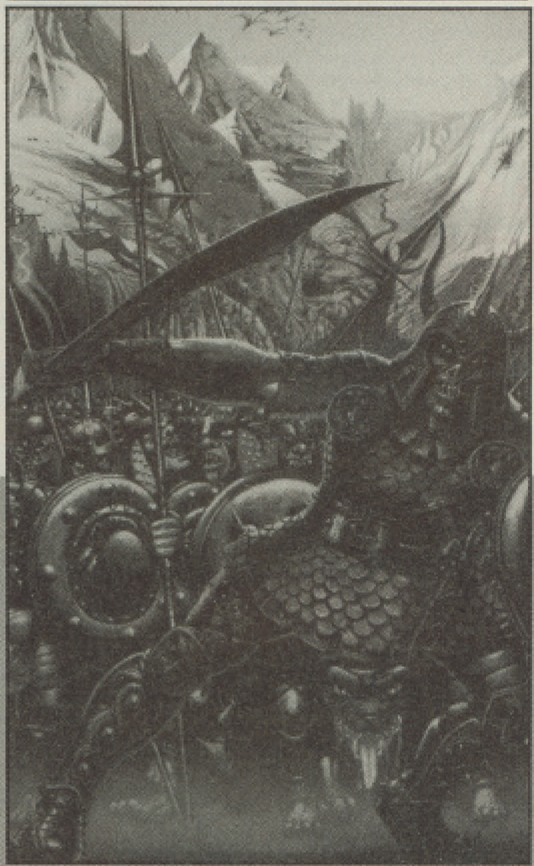
1. Eclipse of the Kai – *Joe Dever & John Grant*
2. The Dark Door Opens – *Joe Dever & John Grant*

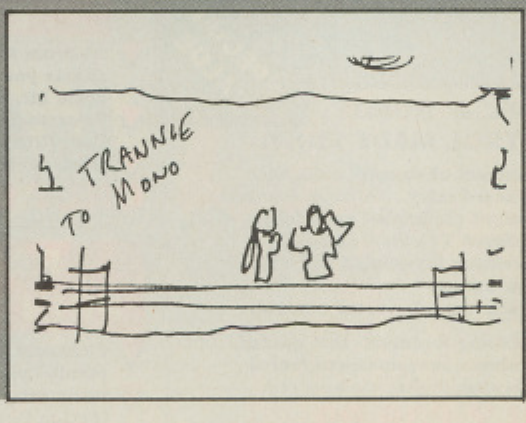
Others

The Magnamund Companion – *Joe Dever & Gary Chalk*

The Lone Wolf Poster Painting Book – *Gary Chalk*







DATEL ELECTRONICS

DATA RECORDER



- Quality Commodore compatible data recorder.
- Pause control, counter, etc.
- Suitable for 64/128.
- Send now for quick delivery.

ONLY £24.99



SPRINT 128

- Full feature Centronics Printer Interface.
- Connect your 64/128 to a range of full size Centronics Parallel Printers.
- Easy to use - supports Commodore Graphics Set.
- Onboard Microprocessor Chip means no power programmes to load.
- Works with most applications.
- No more to buy!

ONLY £29.99

EPROM ERASER

- Erases up to 4 Eproms at once.
- Small size with built-in timer.
- Even fits in your pocket!
- Comes complete - no more to buy.

ONLY £39.99

ROBOTEK 64



SPECIAL VALUE

MODEL & ROBOT CONTROL MADE EASY.

- 4 output channels - each with onboard relay.
- 4 input channels - each fully buffered TTL level sensing.
- Analogue input with full 8 bit conversion.
- Voice input for voice control.
- Software features:- test mode/ analogue measurement/voice activate/digital readout etc.

ONLY £39.99

INCLUDING HARDWARE/SOFTWARE/ MIC. ETC,(STATE TAPE OR DISK).



MIDI 64

FULL MIDI INTERFACE FOR THE 64/128 AT A REALISTIC PRICE.

- MIDI in/MIDI thru/2 x MIDI out.
- Compatible with most leading software packages.

ONLY £29.99



DIGITAL SOUND SAMPLER

- The new sampler allows you to record any sound digitally into memory & then replay it with astounding effects.
- Playback forwards/backwards with echo/reverb/ring modulation.
- Now with full sound editing module to produce outstanding effects.
- Full 8 bit D to A & ADC conversion.
- MIDI compatible with suitable interface. (i.e. Datel unit for £29.99, see ad.).
- Live effects menu includes real time display of waveforms.
- Line in/mic in/line out/feedback controls.
- Powerfull sequencer with editing features.
- Load/save sample. Up to 8 samples in memory at one time.
- Complete software/hardware package. Tape or Disk (please state).

ONLY £49.99

COM-DRUM

GITAL DRUM SYSTEM

- Now you can turn your digital sound sampler into a digital drum system.
- 8 digital drum sounds in memory at one time.
- Complete with 3 drum kits. Real drum sounds - not synthesised.
- Create superb drum rhythms with real & step time.
- Full editing. Menu driven. Load/ Save facilities.
- Output to hi-fi or through TV speaker.

ONLY £9.99

STATE TAPE OR DISK

COM-DRUM EDITOR

- 24 drum sounds supplied on disk to enable you to construct your own drum kit.
- Re-arrange sounds supplied with a Com-Drum to make new kits.
- Load/save facilities.

ONLY £4.99 DISK ONLY

PARALLEL PRINTER CABLE

- Connects full size printers to the parallel port of your C64/128.
- Many programmes and cartridges (Action Replay/Final Cartridge etc.) will drive printers from this port.

ONLY £12.99 COMPLETE

A TOTAL MIDI MUSIC PACKAGE



SAVE OVER £70

UNBEATABLE VALUE PACK - THE YAMAHA SHS 10 FM MIDI COMPATIBLE SHOULDER SYNTHESISER, THE ADVANCED MUSIC STUDIO SOFTWARE & MIDI INTERFACE PLUS FREE MIDI CABLES TO CONNECT EVERYTHING TO YOUR COMPUTER SYSTEM

FREE MIDI CABLES

YAMAHA SHS 10 FM SYNTHESISER KEYBOARD

- Superbly styled guitar-type keyboard with shoulder strap.
- Top quality brandname.
- 2.5 octave keyboard.
- 25 built-in instrument and rhythm choices.
- Uses FM synthesis.
- Full MIDI standard.
- Superbly styled guitar-type keyboard with shoulder strap.
- Requires 6 "C" batteries or AC/DC adapter.

ADVANCED MUSIC SYSTEM

- A full feature Music Studio with MIDI.
- Editing and sequencer features.
- Many features - too many to list!
- Sorry, disk only.

MIDI INTERFACE

- MIDI In/Out/Thru.
- Simply plug in.

FOR ONLY £99.99

NO MORE TO BUY!!

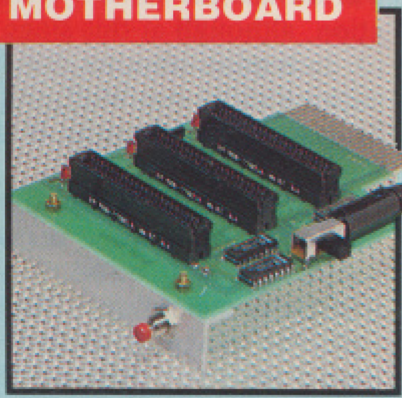


RESET CARTRIDGE

- Unstoppable reset button. NOTE:- user port cheaper type reset buttons offered by others are not unstoppable.
- Resets even so called "unstoppable" programs.
- Add pokes from magazines etc.
- Simply plugs in to cartridge port.

ONLY £5.99

3 SLOT MOTHERBOARD



SAVE WEAR & TEAR ON YOUR EXPANSION PORT

- Will accept three cartridges on it's high grade PCB.
- Switch in/out any slot.
- Fully buffered.
- Reset button and an onboard safety fuse.

ONLY £16.99

TOOLKIT IV

THE ULTIMATE DISK TOOLKIT FOR THE 1540/1541.

- A disk toolkit is an absolute must for the serious disk hacker. Toolkit IV has more features than most for less.
- DISC DOCTOR V2 - Read & write any track & sector including extra & renumbered tracks. Repair damaged sectors.
- HEADER/GAP EDITOR - Decodes & displays ALL header information including off bytes & header gap. Rewrite the entire header & header gap. Renumber sectors. Also edit any sector tail gap.
- DISK LOOK - Sort directory. Recover lost files. Display file start /end addresses. Disassemble any file program directly from the disk to SCREEN or PRINTER including undocumented opcodes. Edit Bam.
- FAST FILE COPY - Selective file copy. Works at up to 6 times normal speed.
- FAST DISK COPY - Copy an entire disk in 2 minutes or less using single 1541.
- FILE COMPACTOR - Can compact machine programs by up to 50%. Save disk space.
- FORMATTER - 10 second format an entire disk or format any individual track or half track 0 to 41.
- ERROR EDIT - Quickly find & recreate all read errors including extras & renumbered tracks or sectors & half tracks from 0 to 41.

ONLY £9.99

DATEL ELECTRONICS

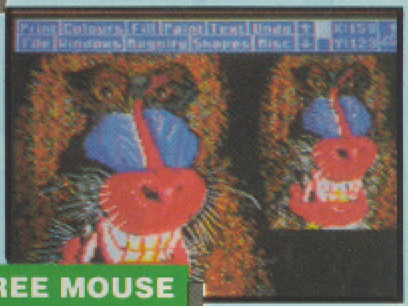
THE ULTIMATE GRAPHICS PACKAGE...



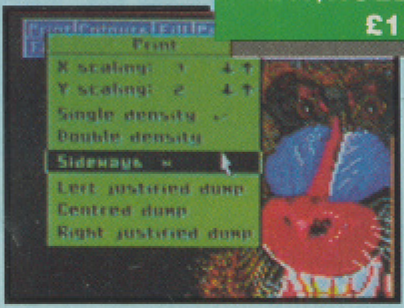
ALSO GEOS COMPATIBLE

SPECIAL OFFER PACK!





INCLUDES FREE MOUSE MAT/HOLDER WORTH £12.99





COMMODORE 1351 MOUSE

COMPLETE WITH

OCP ADVANCED ART STUDIO™

ONLY £29.99

TOTAL PACKAGE INCLUDING 1351 MOUSE/ MOUSE MAT/ HOLDER AND OCP ART STUDIO

- **1351 Mouse is a high resolution two button mouse featuring optical counting, teflon guides, microswitches, rubber coated ball and high quality interface.**
- **When combined with OCP Advanced Art Studio this graphics package is quite simply the best system available. The features are unmatched...**
- Create an image - shrink it, expand it, move it, rotate it, copy it, colour it etc., etc.
- Spray patterns or shades, make elastic lines - stretch and manipulate shapes.
- Zoom in to add detail in fine mode.
- Pulldown/Icon driven menus for ease of use.
- Mouse operation, plus joystick and keyboard control.
- 16 pens, 8 sprays, 16 brushes - so flexible anyone can create superb graphics easily.
- Full cut and paste facilities plus excellent printer support.
- Pixel edit, font editor, flip, invert, rotate, solid or textured fill and professional manual make Art Studio simply the best graphics package for the Commodore (please state Tape or Disk).



VIC 20 RAMPACK

- Simple plug-in memory expansion cartridges for your VIC 20.
- Never be short of memory again!
- Two models available - 16K or 32K.
- Both models are switchable for various memory settings i.e. 3K, 16K, 24K, etc.
- Many VIC programs require extra RAM - this is your answer.

ONLY £14.99 FOR 16K

ONLY £19.99 FOR 32K

NOW YOU CAN INCREASE THE MEMORY OF YOUR C16 EASILY



C16 RAMPACK

- Plug in RAM cartridge gives you up to 16K EXTRA MEMORY!!
- That's over 28K free for Basic and even an extra 18K in HiRes mode!
- Plugs in to memory expansion port - then just switch on.

ONLY £14.99

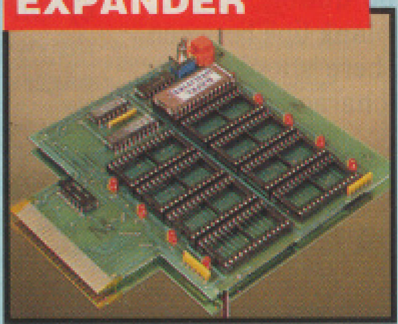
EPROM ERASER

- This handy AC/DC Eprom Eraser will erase up to 4 chips quickly and easily.
- Built-in timer and safety switch on cover.
- Superbly styled - small enough to fit in pocket - even as it works!!
- Works from built-in battery (supplied) or from AC/DC adaptor if required.

ONLY £39.99

LIMITED NUMBERS AT THIS AMAZINGLY LOW PRICE!!

256K SUPEROM EXPANDER



- Select instantly from 8 sockets which accept up to 32K EPROM each.
- Program your own EPROMs using our EPROM programer.
- No need to have lots of cartridges - just make a selection from the Superom menu.
- Fully menu driven on power up.
- Select any slot under software controls.
- Unique EPROM generator feature will take your own programs - basic or m/c & turn them into autostart EPROMs. (EPROM burner required).
- Accepts 2764/27128/27256 EPROMs.
- On board unstoppable reset.
- On board operating systems - no programs to load.

ONLY £29.99

EPROMMER 64™



- A top quality, easy-to-use EPROM programer for the 64/128.
- Fully menu driven software/ hardware package makes programing/reading/verifying/ copying EPROMs simplicity itself.
- Will program 2716, 2764, 27128 & 27256 chips, 12.5, 21 or 25 volts.
- Fits into user port for maximum compatibility with cartridges/ Superom Board etc.
- Full feature system - all functions covered like device check/verify.
- We believe Eprommer 64 is the most comprehensive, most friendly & best value for money programer available for the 64/128.
- Ideal companion for Superom Board, Cartridge Development System, our kernal expanders or indeed any EPROM base project.
- Comes complete with instructions - plus the cartridge handbook.

ONLY £39.99 COMPLETE

ALL ORDERS NORMALLY DESPATCHED WITHIN 48 HRS

HOW TO ORDER ...

BY PHONE	BY POST	FAX
0782 744707 24hr Credit Card Line	Send cheques/POs made payable to "Datel Electronics"	0782 744292 UK ORDERS POST FREE EUROPE ADD £1 OVERSEAS ADD £3

PRICES AND SPECIFICATIONS CORRECT AT TIME OF PRESS AND SUBJECT TO CHANGE WITHOUT NOTICE

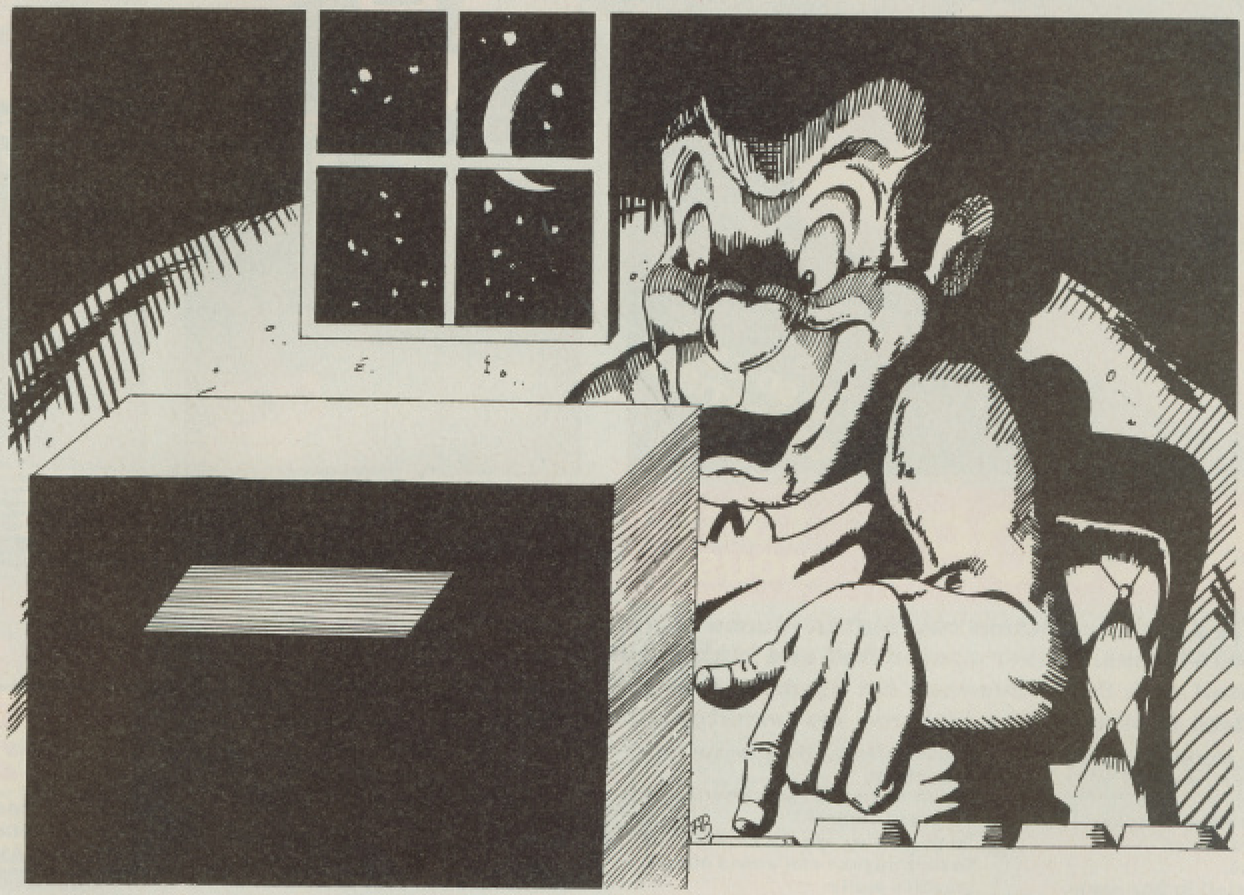
CALLERS WELCOME - Please reserve goods by telephone prior to visit.

DATEL ELECTRONICS

DATEL ELECTRONICS LTD., FENTON INDUSTRIAL ESTATE GOVAN ROAD, FENTON, STOKE-ON-TRENT, ENGLAND.

SALES ONLY	TECHNICAL ONLY
0782 744707	0782 744324

Contributions



So you have written a program? You want it published? Follow these guidelines when you send it in to us!

Your Commodore is always on the lookout for new programs, hints, tips, articles and even regular series. In fact, if you have something that you think could be of use to other Commodore owners, we want to hear about it.

Below you will find a list of guidelines that will help us to deal with any item you send in. We don't expect everybody to be budding William Shakespeares, although if you do follow these simple rules it will make our job a lot easier. (And speedier).

We are constantly striving to make our magazines more professional in their approach. Therefore, it is to this end that we are making some alterations to how programs should be submitted with a view to publication.

In future, whenever you submit programs or articles for possible publication, we ask you to ensure it is submitted according to the guidelines set out below. Please note:-

Any programs NOT submitted along these guidelines will automatically be rejected.

Submission Guidelines

1) If possible all material sent to the magazine should be typed or printed out on a computer printer.

2) All text should be double spaced, ie: there should be a blank line between each line of text. There should be a margin of at least 10 characters around the text.

3) The very first page should consist of the following:

Name of the article/program
Machine that it is for
Any extras needed – drive/printer, etc.
Your name, address and telephone number.

4) The top of every page should have the following information on it:
Program title
Your name
Page number

For example:

Rasters/A.Bloggs/Page 1

5) Do not make additional marks on your text, especially underlining.

6) On the bottom of each page put the word MORE if there are more pages or END if it is the last one.

7) If possible enclose a listing of your program. If written in machine code, then a listing of the source code would be appreciated.

8) Programs should be on either tape or disk with TWO copies included. If problems appear then we have more chance of successful loads. If the program is less than 10 lines it can be included in the text.

9) If your article needs any artwork then supply clear examples of what is needed. We don't expect you to be a first class artist but we do need to see what is required.

10) Photographs, if necessary, must be either black and white prints or colour slides. We can take shots ourselves so don't worry about this too much.

11) Submissions can be of any length. If you have a five line routine or a full six part series, both will be welcome.

12) Payments are normally made 3 to 4 weeks following publication. Payment varies quite a lot and depends on quite a number of factors, such as complexity of program, presentation and number of pages it uses. Payments vary from £10.00 to £800.00 for series.

13) If we find your submission suitable for publication we will write giving you details of terms and payments. Prompt return of the acceptance will ensure the minimum of delay in seeing your article/program appear in print.

14) If you want your material returned, should we find it unsuitable for publication, then a stamped addressed envelope should be included. NOTHING WILL BE RETURNED IF THIS IS NOT INCLUDED.

That just about sums up our requirements for your submissions. One final word... Get writing!!

To Recap – Follow These Guidelines

1) A brief introduction to your program. What it is, what it does and why everyone should use it. Make it sound interesting enough for people to read.

2) A breakdown on how the program works. Included in this general heading will be your program flow. Why you are doing specific tasks. If you are using M/C, follow each routine through, (Not line by line.). Explain your sub-routines if using Basic. In short, we require an indepth explanation on how your code works, be it Basic or M/C.

3) Instructions on using the program. We need clear, concise instructions. For example, if you have written an extended basic, examples need to be shown. If it is an art package, again examples are needed.

4) A getting it in section. If you have written a multiple part program. Or if Basic needs to be shifted around. Indeed if there are any oddities at all, make sure that you give clear instructions on how to input it.

The above sections should be presented either on TYPED plain paper or as a text file using a suitable wordprocessor. Each page should be numbered. The necessary headings and footings should be included, and the documents should have double line spacing. We require two copies of your program. (This is essential when sending programs on disk, they can and do get damaged in transit).

In order to make our magazines more appealing, we feel it is necessary to insist on these guidelines. After all, it is you, the user that will gain most from this. The less work we have to do in preparing your submission, the better the final product will appear.

An Introduction To Plus/4 Machine Language

Mark Everingham reveals the inner workings of his Plus/4

In future installments of *FourGround*, we shall be printing many articles and programs which need a basic understanding of Machine Language programming. For this reason, over the next few months *FourGround* presents a brief beginners course to machine language. It is obviously not possible to present a full discussion of the subject within the limited space of the magazine, but the course aims to provide a foundation on which to build more advanced knowledge.

For most people, the phrase "machine-code" has horrible images attached to it - strings of binary, incomprehensible listings and degree level mathematics. However, the fears the majority have about machine-code are totally unfounded - the language can be easy and extremely rewarding to use. If you have tried to learn machine-code before and given up in frustration, don't run off now, most books on the subject tend to be extremely badly written, full of jargon and irrelevant information. This short course has been written to be compact and easy to understand while retaining all the essential information. For those who can already program in machine language, it should serve as a useful reference work.

What Is Machine-Code?

"Machine-code', "Machine Language" or "Assembly Language" is a computer language just like any other - Basic, PASCAL or C. Machine Language is the native language of the Plus/4 and in fact the computer at its most basic level cannot understand anything else. Basic may seem like the computers natural language, but is really nothing more than a program written in machine language. At the heart of your Plus/4 is a silicon chip called the 7501 Central Processing Unit (CPU). When you write programs in machine language, you are sending instructions directly to this chip. When you write in Basic, your Basic programs have to be converted into machine-code as they are run, because the 7501 cannot understand them in their Basic form.

So why program in machine language? The reason most people use Basic is because it is so easy to learn. For example, to draw a circle in Basic takes just one command. In machine language, hundreds of individual commands are needed. This might make Basic seem far more attractive, but the problem with Basic is that it runs incredibly slowly. Machine language runs commands at up to a million instructions per second, whereas with Basic you're lucky if you achieve 60 per second. Another advantage of machine language is that because you're working at such close proximity to the actual hardware of the machine, you have the ultimate in power over every individual function that the Plus/4 hardware can perform. Once you have begun programming in machine language, Basic will, I can assure you, seem horribly inefficient.

Number Bases - Decimal, Binary, and the Dreaded Hex!

Computers are number machines - Your Plus/4 computer knows nothing about anything but numbers. Everything it does concerns numbers, even the commands it executes are stored as numbers. We human-beings tend to use the number base called decimal or denary, simply because we happen to have ten fingers on our hands, but computers don't like decimal, so instead they use "Hexadecimal" or "Binary".

The concept of number bases is really simple. Think of a decimal number, say 12345. The key to understanding the magnitude of the number is column headings. Look at the number 12345 written under column headings below.

Heading:	10000	1000	100	10	1
Value:	1	2	3	4	5

In order to work out what number is written down, we just add up the products of the values and headings: 1x10000+2x1000+3x100+4x10+5. Each column heading is a power of 10, so the first one is 1, (10∧0), the second one 10x10, the third 10x10x10 and so on. The principle for hex and binary is identical, except that for hex, the base is 16 and for binary it is 2. This means that the column headings are 1, 16, 16x16x16 etc. for hex, and 1, 2, 2x2, 2x2x2 etc. for binary. To convert a number to decimal, just repeat the process above using the new column headings. For binary, this is easy, as each column can only hold a number 0 or 1. The number 170 is shown here:-

Heading:	128	64	32	16	8	4	2	1
Value:	1	0	1	0	1	0	1	0

The final value is calculated as 1x128+ 1x32 + 1x8 + 1x2. The same method is applied for hexadecimal, except that if the base is 16, each column needs to hold a number between 0 and 15, whereas in decimal only between 0 and 9. This means we need 6 more digits. For these we use the first six letters of the alphabet, A=10, B=11, C=12... F=15. The number 4660 is shown below:-

Heading	4096	256	16	1
Value:	1	2	3	4

Of course, if in a program we used some numbers which were binary, some decimal and some hex, we couldn't tell the difference – the number 101 could be 101 decimal, 5 binary or 257 hex. The answer is that binary numbers are preceded by a "%" (percent) sign, hex by a "$" (String/Dollar) sign, and decimal numbers have no prefix. This avoids all possibility of confusion. For most of the time, however, you will find you use hexadecimal, so the problem need not arise.

So why do we use hex and binary instead of decimal? Computers are electronic devices, made up of banks of transistor switches. A switch can only be On or Off, and these settings can be effectively represented by 1 and 0. These 1's or 0's are called bits – Binary Digits referred to by their column headings 0-7 (right-left), and collections of eight of them (255 decimal) are called Bytes. Hexadecimal is used because it is much shorter to write down, while retaining a similarity to binary. The number 255 in decimal needs three digits, in binary, 8, but in hex it needs only two.

The Importance Of Numbers

So far we have only talked about numbers for numbers sake, and this is the criminal error which most machine-code tutors make. In machine language, nothing but numbers are used, but the important thing to remember is that a number can mean anything. In real life, numbers can be house numbers, telephone numbers, prices, times, or a host of other things. Machine language is identical, numbers can be colours, sounds, commands, speeds or anything you wish. All of the commands discussed in the section "The Instruction Set" perform operations on numbers. If you can remember that these numbers can be of any significance, machine-code programming will be elementary!

The Memory Machine

Your Plus/4 is described as a 64K computer. This means that it has enough memory chips to store 65535 byte long binary numbers. You might expect 64x1000, but 65535 is equivalent to 64x1024, 1000 being near enough to 1024 which is a nicer binary number. Because of the binary system, each of these 65535 "locations" can hold a number in the range 0-255, and each location is referenced by an "address". This is simply a number between 0 and 65535 which points to a byte of memory. In addition, your Plus/4 has two types of memory: ROM (Read Only Memory) and RAM (Random Access Memory). The ROM holds the machine-code program from BASIC. It lives at address $8000-$FFFF, and cannot be changed. The address space $0000-$7FFF is the RAM, although the extra 32k is found at $8000-$FFFF, under the ROM. For the moment we will only consider the lower 32k.

Each byte of RAM can be put to a different use. To find out which byte of RAM is used for what, you should consult a manual such as ANCO's "C16-PLUS/4 REFERENCE BOOK" which has a full list. Many locations are used for things like the screen, Basic calculations or Basic programs. When you write a program in Basic, the Basic program in ROM finds a place in RAM to store the program and puts it there. In machine language, you have to tell the Plus/4 where to put the program, and you must take care not to put it in a place which is used by another part of the computer!

The Tedmon Machine-Code Monitor

On almost all computers, writing a program in machine-code means that you have to go out and buy a piece of software called an Assembler, luckily for you, the Plus/4 has a simple one built in, called TEDMON.

Machine-code programs are written like Basic as a series of text commands, such as "TXA", "LDA $8000,X" and "BRK". Each command is represented by a three-letter mnemonic, or abbreviation. To actually run these commands, they must be converted into a series of numbers and stored in RAM. This process is called Assembling.

The TEDMON program on ROM in your Plus/4 will carry out the simple assembly process for you to save you doing all the work. To assemble a program, type "*MONITOR*" from Basic. In order to assemble a program, you have to tell Basic where to put the program. Consult a memory map if you have one to find a suitable place, but for most of the time, a good place to put programs is at $6000. This location in memory leaves you with 8k free for Basic even with a graphics screen in operation, leaving 8k for your machine-code program – an amount you are not likely to use up for a long, long time! The TEDMON assemble command takes the below form:

```
A $aaaa xxx operand
```

The "A" is the actual assemble command, and can be replaced by a "." (period) if you prefer. "$aaaa" is the address at which you wish the line of machine-code to be stored. It should always be expressed in hex and preceded by a dollar symbol. "xxx" is the three-character mnemonic of the command you wish to be assembled, and "operand" is the operand or parameters the command takes. When you enter a line and push (Return), the line is stored in memory, and a new address is printed, allowing you to enter the next line. All numbers entered into TEDMON should be hexadecimal, and preceded by a dollar symbol.

When you have finished entering a program, delete the address on the screen. You can exit to Basic by pushing "X" and (Return) for exit, or you can run your program by typing "G" plus the hex start address of your program. The dollar symbol should not be entered.

Once your program is in memory, you can look at it or edit it by typing "D" plus the address at which you want to start looking. A series of lines will be displayed on the screen. This is called Disassembly, and the lines on the screen can then be edited and re-entered using the normal Screen Editor functions.

We have not covered here every function of TEDMON, but only the most important ones. Full coverage of the TEDMON monitor can be found in your Plus/4 manual.

7501 Processor Registers

As I have said, all of machine-code's commands operate

on numbers. These numbers can be held in three places – in a memory location, in the memory directly after the machine-code command, or in one of the 7501 registers, which are like fast, internal bytes of memory.

The Program Counter

The program counter (PC) is the only 7501 16-bit register. In it is stored the address where the next command to be executed can be found. When an instruction is executed, the PC is automatically incremented so that it points to the address of the next instruction. When you direct control to another location using a JMP, JSR or branch instruction (see "The 7501 Instruction Set"), the address in the PC is changed accordingly, so you don't have to worry about it.

The Status Register

The status register (SR) is an 8-bit (byte) register which is used to indicate the states of various options of the 7501 CPU. In fact, only seven of the eight bits are actually used, and the functions of these bigs are discussed below. Each bit is known as a "flag".

Bit 7 – The N (Negative) Flag. The negative or N bit is set to 1 when the last numeric result had bit 7 set, and is set to 0 when the number had bit 7 cleared. Effectively, numbers over 127 set the N bit, and those below clear it. If bit 7 is used to show the sign of the number, the N bit reflects this sign accordingly.

Bit 6 – The V (Overflow) Flag. The Overflow (V) bit is set to 1 when the last operation resulted in what is known as a two's complement overflow. This is only of use in signed arithmetic, where bit 7 represents the sign (1=-, 0=+). Overflow is set when the status of bit seven has changed incorrectly, in such cases as two numbers yielding a result greater than will fit in one sighed byte.

Bit 4 – The B (Break) Flag. The Break bit is set to 1 when the last operation was a "BRK", or is cleared if the operation was anything else.

Bit 3 – The D (Decimal) Flag. Setting the decimal bit enters the decimal mode of the 7501 processor. Clearing it resorts to binary arithmetic. For information on the decimal mode, see the SED instruction.

Bit 2 – The I (Interrupt) Flag. Setting the interrupt (I) bit to one causes interrupts to be disabled, or clearing enables interrupts. The function only works with IRQ (Interrupt ReQuest) interrupts, and is covered fully in the section on interrupts.

Bit 1 – The Z (Zerp) Flag. The zero bit is set to 1 when the result of the last operation was zero, for instance when subtracting a number from itself. If the result was any number but zero, the zero bit is cleared.

Bit 0 – The C (Carry) Flag. The carry bit is set to 1 when the last addition gave a result which could not be expressed in 8 bits. When subtracting, it is cleared if a borrow out of eight bits was required. The bit is also used by the shift and rotate instructions, where the last bit is shifted into the carry flag.

The Accumulator (.A)

The accumulator, or .A register is the most important of the 7501 registers. It is an 8-bit register which is used for all arithmetic and bit manipulation operations, as well as being used for most data transfer functions.

The X Index Register (.X)

The .X register is one of a pair of "index" registers. It is like the accumulator, in that it is an 8-bit register, but only the most simple operations can be carried out on .X. Its primary use is as an index for indexed addressing modes – See the section on addressing modes.

The Y Index Register (.Y)

The .Y register is identical to the .X register, also being used mainly as an index.

The Stack Pointer (SP)

The stack pointer is another 8-bit register, which contains a pointer to the current byte on the 7501 machine stack. See the section which follows on the stack for an outline of its use.

The 7501 Processor Stack

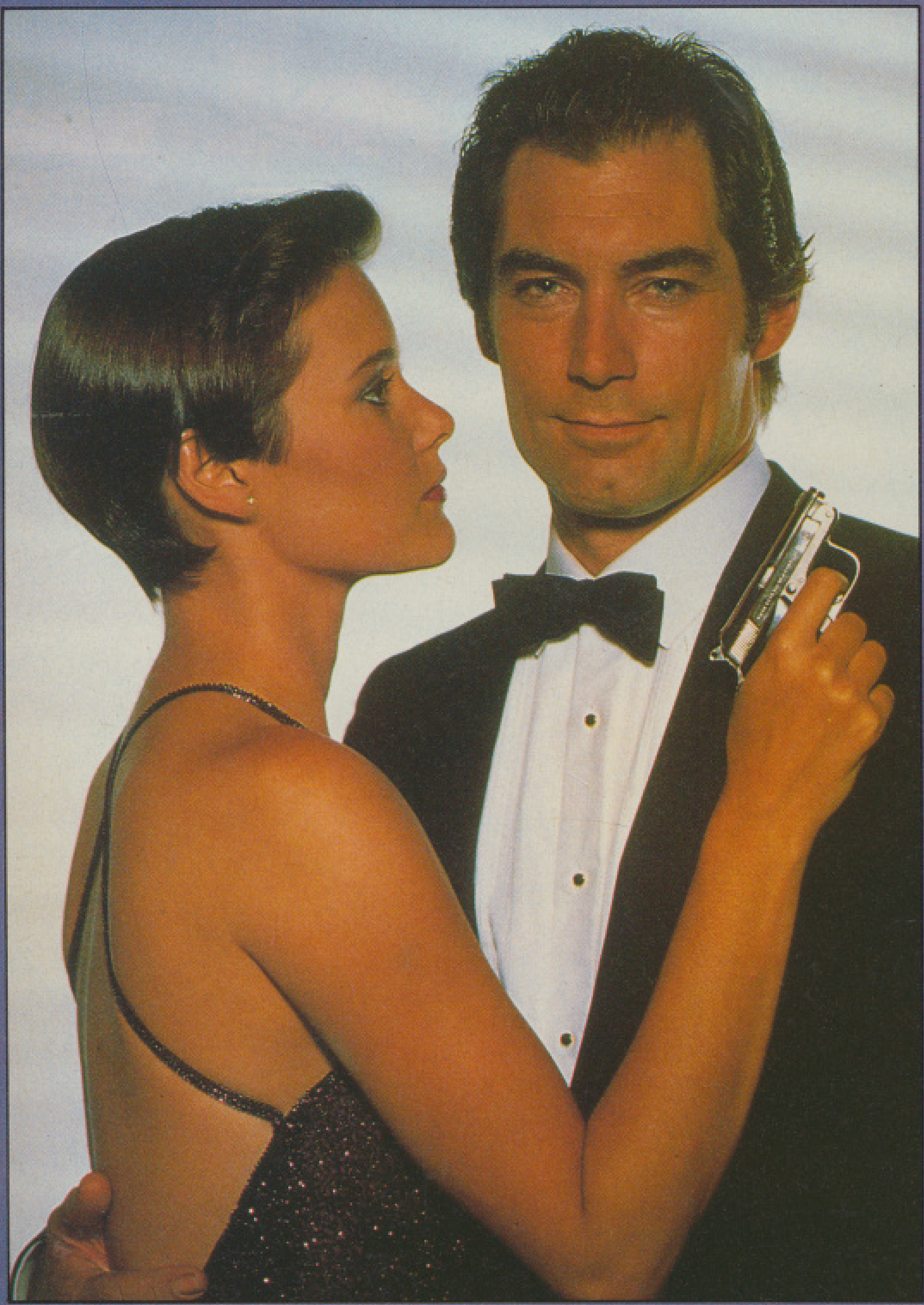
In Basic, you have two instructions to alter the flow of a program – GOTO and GOSUB is simple – it just performs an absolute, unconditional jump to a line. GOSUB does the same thing except that before the jump is performed, the statement to which the program should return is first pushed into an area of memory called the Basic stack. Then, when a RETURN statement is encountered, the program pulls the return statement back off the Basic stack and jumps to it. machine language works in the same way using its own machine stack, except that the GOSUB command is replaced by JSR (Jump to Sub-Routine). When JSR is executed, the address of the next command is pushed onto the stack so that when a subsequent RTS (ReTurn from Subroutine) is found, the address can be pulled back off the stck and transferred to the PC to continue program execution. The process of pushing and pulling is automatic and controlled by the Stack Pointer. The machine stack is in fact a 256-byte (1 page) area of memory located at $0100 onwards.

In addition to being used for sub-routines, the stack can also be used as a temporary store for numbers, avoiding the need to use a specific area of memory. The PHA, PLA, PHP, PLP instructions push numbers onto the stack and pull them back of it again. When a number is pushed or pulled, the stack pointer is decremented or incremented accordingly. You can however control the stack pointer contents directly using the TXS and TSX instructions. The stack pointer is a single 8-bit index into the stack area at $0100.

Note that when using the stack, you must of course match every JSR with an equivalent RTS, and push with equivalent pull. The stack can accomodate 256 bytes, so upto 128 levels of nested sub-routines are theoretically possible. It is extremely unlikely that you will ever need more than this.

We've had to cut Mark short here, but you'll find more in the second installment.

A Serious Case of Bondage



Crosby... Kevin Crosby takes a look at the latest Bond film and its accompanying game from Domark. After 27 years the world's best known secret agent – James Bond returns once again and, as the film posters state, "This time he's out for revenge".

Licence to Kill is the 16th official Bond movie (not counting Casino Royale and Never Say Never Again) and the second for new boy Timothy Dalton (not the last as some television networks would have us believe!) Dalton himself is very pleased with 'Licence'. "It's much closer to the original character in the books" he enthused at the recent premier in London's Leicester Square Theatre. "It's tough, tense, exciting, everything a Bond film should be". It's certainly a lot tougher than previous films which has been the main reason for Licence to Kill's 15 certificate (the first time a Bond film has received anything other than a 'PG' or equivalent). This was only obtained after over a minute of the more violent moments were cut.

However, this doesn't detract from the film's enjoyment, which is the main thing really. It's just a shame that our film censors in this country aren't a little more like their counterparts on the other side of the Atlantic who have given it a PG13, this means that anyone can see it with a parent but you have to be 13 or over to see it on your own. Seems reasonable enough to me!

The Film itself centres around the twilight world of Drugs Baron Frank Sanchez – an unsavoury character who escapes imprisonment and gets his own back on those that caught him (Bond and Felix Leiter). Now Bond becomes a vigilante to get Sanchez back for what he's done, but he must do it without the backing of her Majesty's Secret Service. This time he's on his own.

All the familiar elements are there; Evil baddies, Beautiful Girls, spectacular stunt sequences and of course those ingenious gadgets courtesy of 'Q', played for the 14th time (he wasn't in Dr No or Live and Let Die) by Desmond Llewellyn. However all these elements have a much harder edge. One could almost be mistaken for thinking this was a Dirty Harry





film, particularly if one is used to the Roger Moore style of Bondry. If that's what you've been brought up on then prepare yourself for a shock... for this is a Bond for the 90's. The action is that bit more violent, the villains are that much more realistic (which makes them all the more sinister) and the Bond girls aren't just the stereotypical Bimbos of the past, they now have quite an effect over the action. Is it welcome? Most certainly. As Dalton puts it "this is an Action, Adventure Thriller for adults that kids can also enjoy". A well written plot from the pens of Michael G. Wilson and Richard Maibaum, competently directed by John Glenn (his fifth Bond Directorship), makes for an excellent entertainment.

It is nice to see Domark has pulled out all the stops to release the game to coincide with the general release of the film. This will be Domark's fourth game based on a Bond film. The first was View to a Kill back in '85, followed by The Living Daylights and Live and Let Die. It'll be launched across all formats, including the Commodore 64, Amiga and PC formats, simultaneously. All three versions of interest to Commodore owners look and play really well, particularly the Amiga version.

The game comprises six parts, split up among three scenes from the film. The first involves James and Felix flying a helicopter at low level over Cray Cay. Avoiding tall buildings and gun emplacements – a little like Thunder Blade. Once through that, you (as Bond) set off on foot after Sanchez but you have to get through plenty of his henchmen first. Weaving your way through a dangerous maze of exploding oil barrels and bullets, you must get back to Felix in one piece. Then it's back in the air to capture Sanchez's plane. Here you are lowered onto the back of the plane where you must attach a towrope and bring Sanchez in.

In Scene Two 007 must foil a narcotics drop, but first he must survive in the water armed with nothing but a knife against divers and armoured boats. Bonuses are added for destroying the Drugs caches as well as escaping with your life. Next, you attach yourself, via Harpoon to a seaplane's pontoons for a high speed, barefoot waterski chase, avoiding catamarans whilst you work your way to the seaplane and gain control.

In the final scenes you are in hot pursuit of Sanchez having just obli-



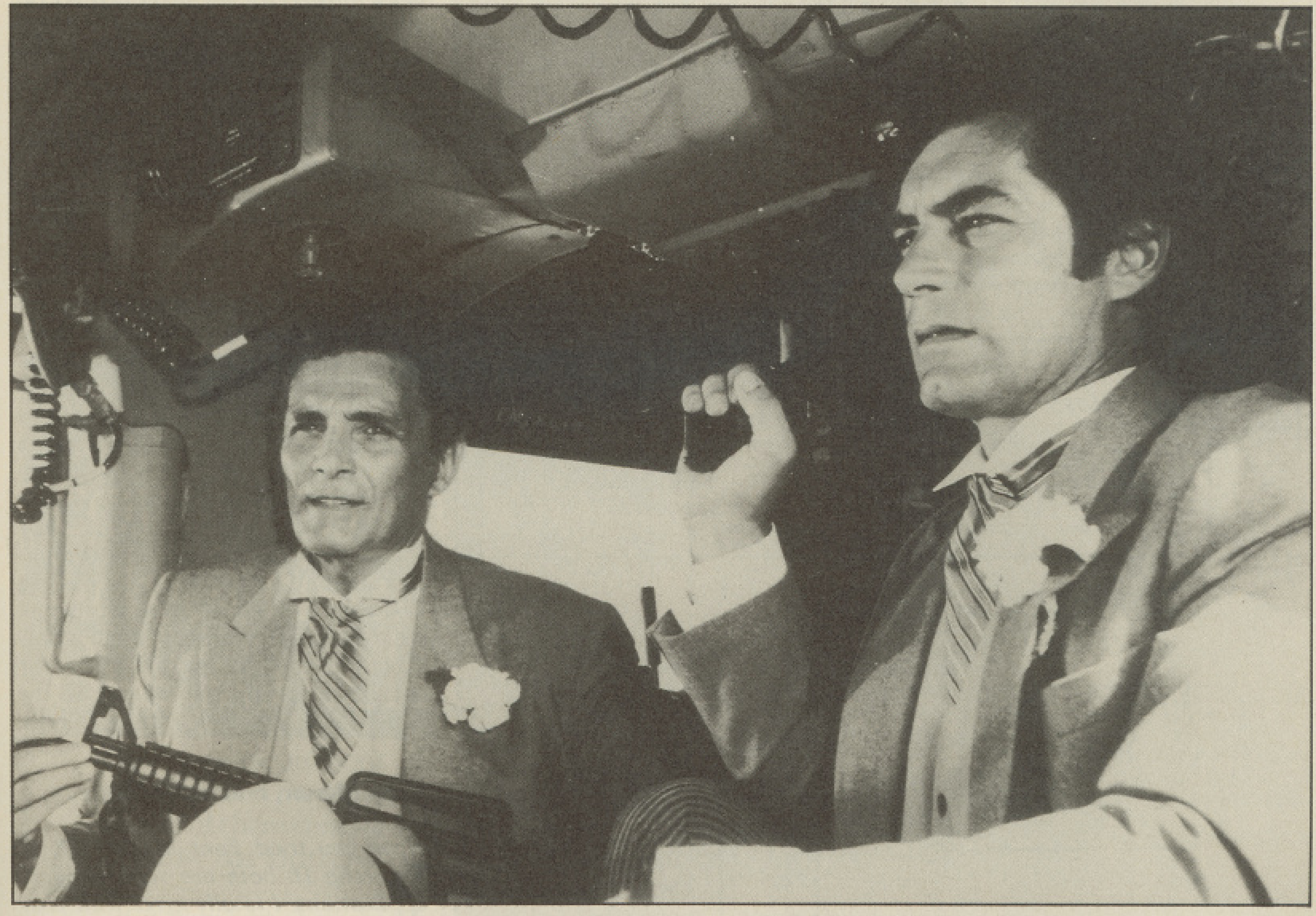


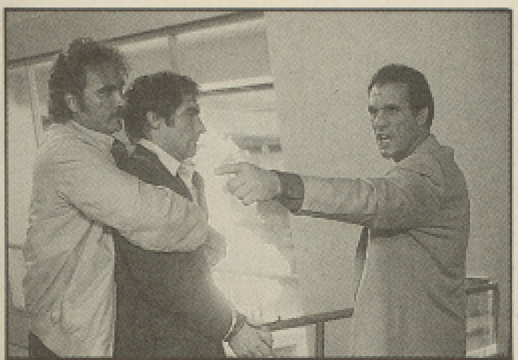
terated his processing plant. He's taken to the winding mountain paths as part of a convoy of 18 wheel juggernauts. The end is near, the focus of your vengeance is just around that bend. You just have to destroy each of the tankers one at a time until you get to the one at the front of the convoy with you know who in it, but beware: he's packing some serious heat in the form of Stinger missiles.

The whole game takes the form of a vertically scrolling, birds-eye view shoot 'em up and works very well as a game. Which as a movie tie-in is rare indeed.

Bond Filmography

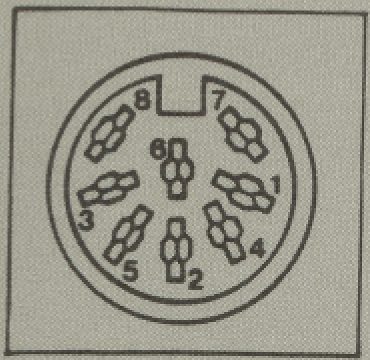
Dr No – (1962) Sean Connery – Dir Terence Young.
From Russia With Love – (1963) Sean Connery – Dir Terence Young.
Goldfinger – (1964) Sean Connery – Dir Guy Hamilton.
Thunderball – (1965) Sean Connery – Dir Terence Young.
You Only Live Twice – (1967) Sean Connery – Dir Lewis Gilbert.
On Her Majesty's Secret Service – (1969) George Lazenby – Dir Peter Hunt.
Diamonds Are Forever – (1971) Sean Connery – Dir Guy Hamilton.
Live and Let Die – (1973) Roger Moore – Dir Guy Hamilton.
The Man With The Golden Gun – (1975) Roger Moore – Dir Guy Hamilton.
The Spy Who Loved Me – (1977) Roger Moore – Dir Lewis Gilbert.
Moonraker – (1979) Roger Moore – Dir Lewis Gilbert.
For Your Eyes Only – (1981) Roger Moore – Dir John Glen.
Octopussy – (1983) Roger Moore – Dir John Glen.
View To A Kill – (1985) Roger Moore – Dir John Glen.
The Living Daylights – (1987) Timothy Dalton – Dir John Glen.
Licence To Kill – (1989) Timothy Dalton – Dir John Glen.



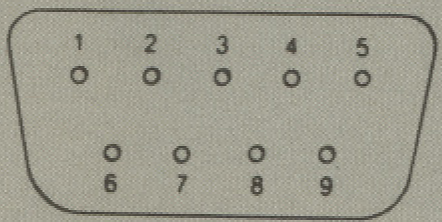


007 LICENCE TO KILL

128 Corner



Pin	Type	Note
1	LUM/SYNC	Luminance/SYNC output
2	GND	
3	AUDIO OUT	
4	VIDEO OUT	Composite signal output
5	AUDIO IN	
6	COLOR OUT	Chroma signal output
7	NC	No connection
8	NC	No connection



Pin	Signal
1	Ground
2	Ground
3	Red
4	Green
5	Blue
6	Intensity
7	Monochrome
8	Horizontal Sync
9	Vertical Sync

Dear 128 Corner
Yes, I own a Commodore 128 and have done since I upgraded from the good old '64 back in 1986. I was pleased to see that you have decided to provide a forum in your magazine for this excellent machine.

I agree with your views regarding the poor availability of software. I suppose the reason for this is that the machine is too small to be a fully fledged business machine and too large to be a game machine. However, by scouring the magazines it is possible to get hold of some magnificent software that is specific to the computer.

I was surprised that you missed out the C128 version of *GEOWRITE WORKSHOP* as an example of a reasonable word-processor. I was initially put off by the *GEOS* system because it did not allow printing in NLQ mode. The fonts that were provided were too 'dotty' on my Star SG10-C for serious letters. This has been solved in *GEOWRITE WORKSHOP.*

The biggest advantage of the program is that you can take large sections of figures from *GEOCALC 128* and paste them into letters without having to write them out again.

For straightforward word-processing I would think that *Superscript 128* is best. Its sister program *Superbase 128* is another program worth seeking out if you want to create a database for any reason. The Basic like programming language, included in the program, allows you to manipulate the data in your own applications programs and adds a new dimension to using a database.

The hype surrounding the *Amiga* makes everyone wonder if they shouldn't part with their old computers and buy a new one. I like playing C64 games on the 128 and am happy with spreadsheet, database and word-processor applications that I have for the machine. In the words of the Americans, if it isn't bust, don't fix it.

Colin Mercer, Bolton, Lancs

Have you tried Font Master 128 or Paperclip II, both are very excellent applications. I totally agree with your sentiments about the 128. I too would like my own personal Amiga, but why buy a Bentley to learn to drive when a Morris Minor will do the trick???

Dear 128 Corner,
It's good to see that you intend to continue covering the C128 computer. I've had one for a few years now and use it quite regularly for wordprocessing and filing. I must say that I prefer using the C128 over the 'real' computers that we have in the office, it's a

lot easier to use and more friendly (I can also play games on it when I'm bored).

When I first purchased my C128 I was very disappointed to find that I couldn't use the 80-column video mode without purchasing an expensive colour monitor, which I couldn't afford. I was intrigued however when I saw people advertising switches that allowed you to use 80 column mode on a cheap monochrome monitor – not as good as colour but certainly better than using 40-columns all of the time. I investigated this further and careful examination of the 80 column video socket in the manual revealed that there is an 80 column composite video output on pin 7 of the connector. Connecting a suitable plug to this pin and the earth (pins 1 and 2) means that the computer can be plugged into a suitable composite monitor.

If you want to use the same monitor to display 40 column video then this can also be achieved. Pin 4 on the 40-column video connector is the video out and pin 2 is ground. Connect the video and ground to a suitable connector as before and you can use the monitor to display 40 columns as well.

I know of a number of other people who use a composite monitor with a lead as described above, in some cases they use pin 1 (LUM/SYNC) instead of pin 4 though on my monitor the results aren't as good.

I have taken my lead even further and placed a double throw switch in to the lead. One pole of the switch goes to the 80-column video pin, the other pole the 40-column video pin. The central pin of the switch is wired to the video plug for my monitor and all of the earth pins are connected together. Switching between 40 and 80 columns is now a simple matter of pressing the 40/80 switch on my monitor and flicking my new switch.

One further advantage of using a video lead such as the one described above is that you can, contrary to popular belief, get 80 columns on your television, as long as you have a video recorder anyway. Most video recorders have an input for a video camera. This input is a composite video one. Plug the lead into this socket, make sure that your TV and video are set up to display from the camera input and you can use your C128 in 80-column mode on your TV. The quality may not be as good as a monitor, but at least it's 80-columns!

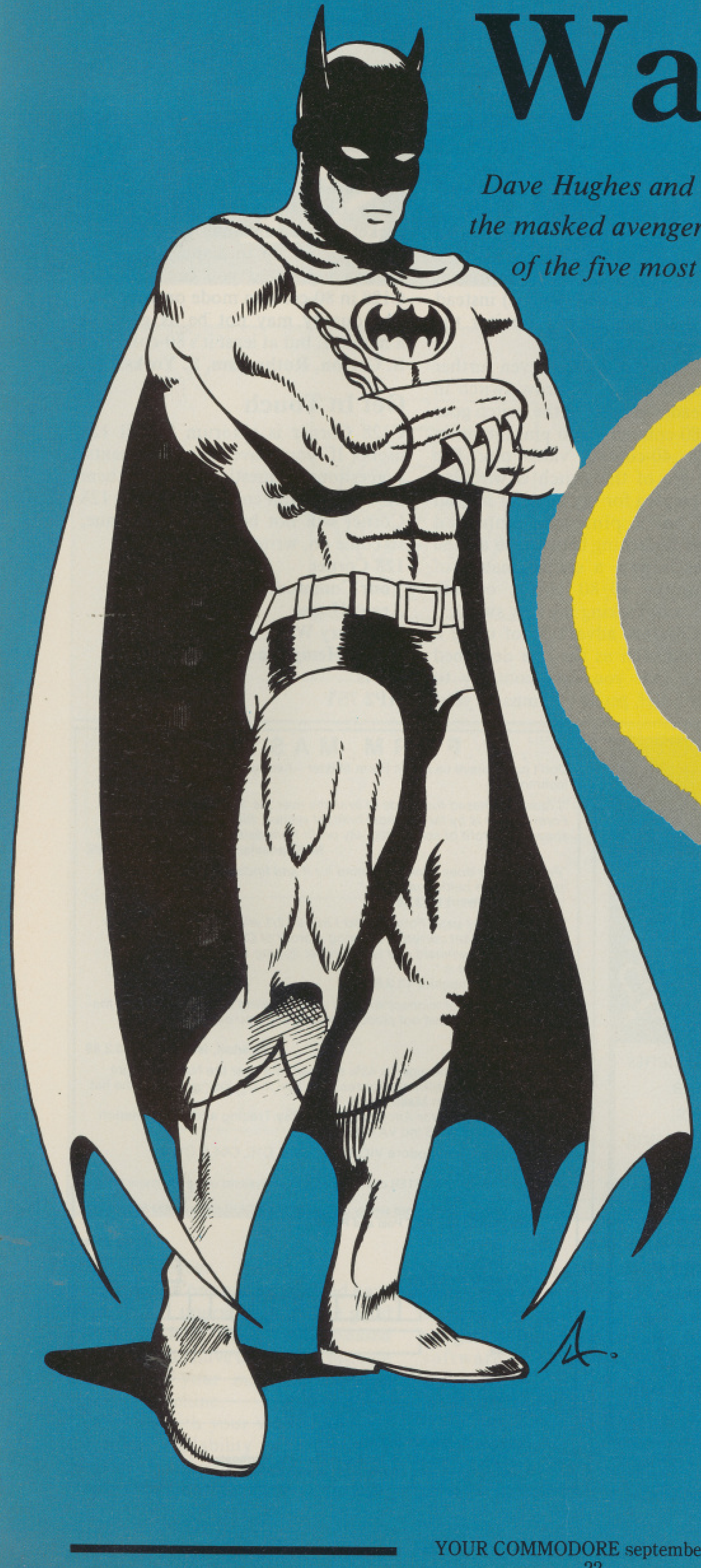
S. Garton, Rotherham, S. Yorks.

Get In Touch

C128 Corner is a forum for all 128 users. If you have any comments, suggestions or questions do send them in. Without your contribution then 128 Corner will not be able to continue, so come on, write to:

128 Corner
Your Commodore
Argus House
Boundary Way
Hemel Hempstead
Herts
HP2 7ST

Waiting fo



Dave Hughes and James Sewell explore the world of the masked avenger, who is considered by some as one of the five most recognised figures in the world.

The Batman, the Dark Knight, the Caped Crusader – call him what you will, his fictional presence has played a major role in our subculture since his conception in a 1939 issue of Detective Comics. Fifty years on the masked vigilante, cloaked and hidden in the dark shadows of a fake New York metropolis known as Gotham City, is about to embark on his second big screen adventure. And this time it's for real.

It is ten years since Warner Brothers, movie-making parent company of Batman's publisher DC Comics, first considered the idea of a new Batman feature film. Since that time, comics like Watchmen and The Dark Knight Returns have brought a new era to the graphic story genre, adding political, racial, sexual and social issues to a field previously associated with muscle-bound, costume-clad freaks of nature. And Batman: The Movie is suddenly big business. Unlike the Caped Crusader's previous screen image, Sam Hamm's original 1986 screenplay concentrated on keeping Batman true to the comics – like Bob Kane's original character, he is a dark brooding vigilante forever seeking vengeance for his dead parents, rather than the ever-so-nice Adam West/Burt Ward team of the camp mid-Sixties spoof series. This time, POW! and ZAP! are left redundant by a movie that pulls no punches.

For this movie, Batman trades in his blue-and-grey leotard for a matt black rubber outfit redolent more of Frank Miller's Batman than Adam West's. This time, no-one will be left smiling at the pranks of his arch-enemy the Joker. And we all know that Batman's boyish sidekick Robin isn't even in the picture – due mainly to the expansion of the part of the Joker which followed Jack Nicholson's reluctant acceptance of the role ("It's like God created him for the role" producer Mike Uslan is quoted as saying). Beetlejuice director Tim

or a Dark Knight

Burton directs, and Beetlejuice himself, actor Michael Keaton, fills the Batman's rubber-clad shoes. Aside from Jack Nicholson, Jerry Hall and Kim Basinger both appear, but Robin the Boy Wonder is not set to appear until BATMAN 2 (already being scripted), about which movie rumours are already flying, including the possible casting of Robin Williams as rhyming rival the Riddler... Originally, however, Sylvester Stallone and Arnold Schwarzenegger were considered for the title role, both being rejected because they did not fit the image of Batman's millionaire alter-ego Bruce Wayne. Also, as Tim Burton says, "the idea of Arnold Schwarzenegger in a Batman costume is frankly ridiculous!". Instead, Michael Keaton wears built-up shoes, a latex chin and false hair inside his padded suit, further enhancing his transformation from playboy Wayne to Dark Knight...

The plot, like the cast list, has also undergone major surgery. Sam Hamm's first draft featured The Flying Graysons and their young offspring (Robin) who, like the young Bruce Wayne, witnesses his parents' murder, this time at the white gloved hands of the Joker. The friendly Sixties image of the Batmobile has been replaced by a Gothic monstrosity which wields a machine gun and is powered by a jet turbine – a fast and powerful reminder of the Batman's might. Gotham City, depicted in the new film at its bicentennial celebrations, is a grimier, darker vision of New York: Commissioner Gordon is fighting police corruption as well as the Joker-dominated underworld collaboration, and the street is filled with the dark and sinister images of wrecked cars, air vents and drug pushers, invading the well-to-do areas and making for a violent and dark base for Batman's comeback.

The Joker is depicted as a rictus-grinning madman, driven insane by his own disfigurement following a chemical accident, and wreaking his vengeance on humanity. Batman is the only thing that stands between him and his evil destiny, and the film follows the interaction between two society misfits, each infatuated with his role and each closer to the other than either would care to admit – a plotline linked closely with British writer Alan Moore's recent novelette The Killing Joke. Closer still to the movie's plot is Frank Miller's highly-acclaimed The Dark Knight Returns, but a crucial scene was (in true Hollywood style) cut from the movie following news that Warner had agreed to a sequel: in the climactic confrontation of Frank Miller's story, Batman only just stops short of throttling the Joker to death, at which the Joker, in his last and best punchline, breaks his own neck so that Batman will take the rap for his death: this scene was written out following Jack Nicholson's agreement to appear in a sequel... The hugh Gotham City set – the largest film set since Cleopatra – stands empty at Pinewood Studios in Buckinghamshire, waiting for the crew to return in February for more adventures of the caped crusader. The movie's success is already secured.

At the time of writing BATMAN is two weeks away from its U.S. release, but already scores of people have gotten bat-fever. Fans have been reported as seeing THE DEAD POOL just to catch the ninety-second trailer for BATMAN shown along with it. Despite Jack Nicholson's opinion that the movie is "doomed", the media coverage and the secrecy in which the movie has been shrouded have made a sure-fire winner for Warner. The Batman comic series The Cult has rocketed to 2,500% of its cover price virtually overnight. Illegal stills, scripts and unfeasibly shoddy pirate video copies are changing hands at massive prices. The merchandising from the series is likely to cover the cost of the $30 million movie in a matter of months, with Warner mounting the biggest merchandising campaign since STAR WARS, with novels, action figures, masks, model kits and even credit cards aping the Batmania of the Sixties.

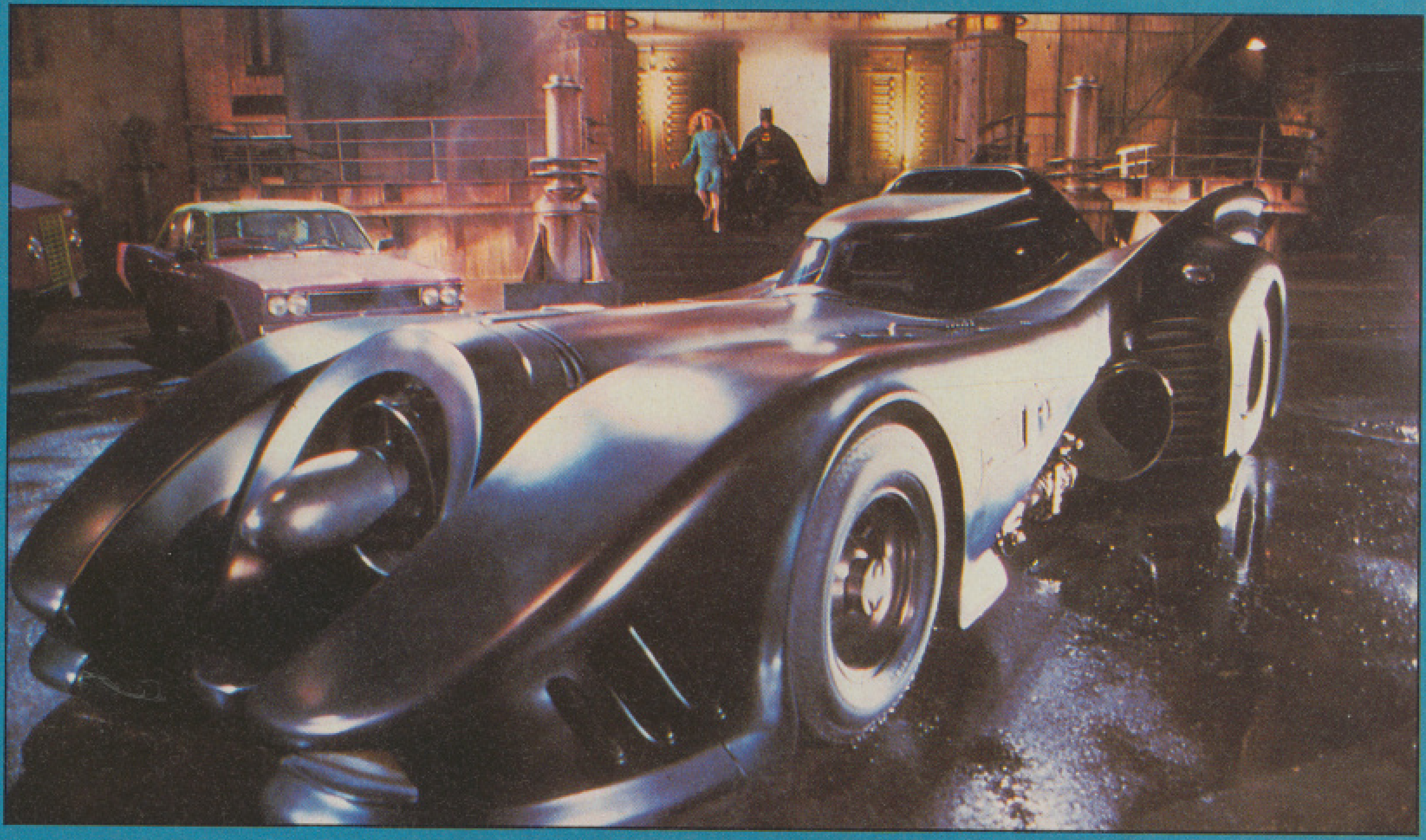
So what is it about this masked hero that demands such slavish devo-

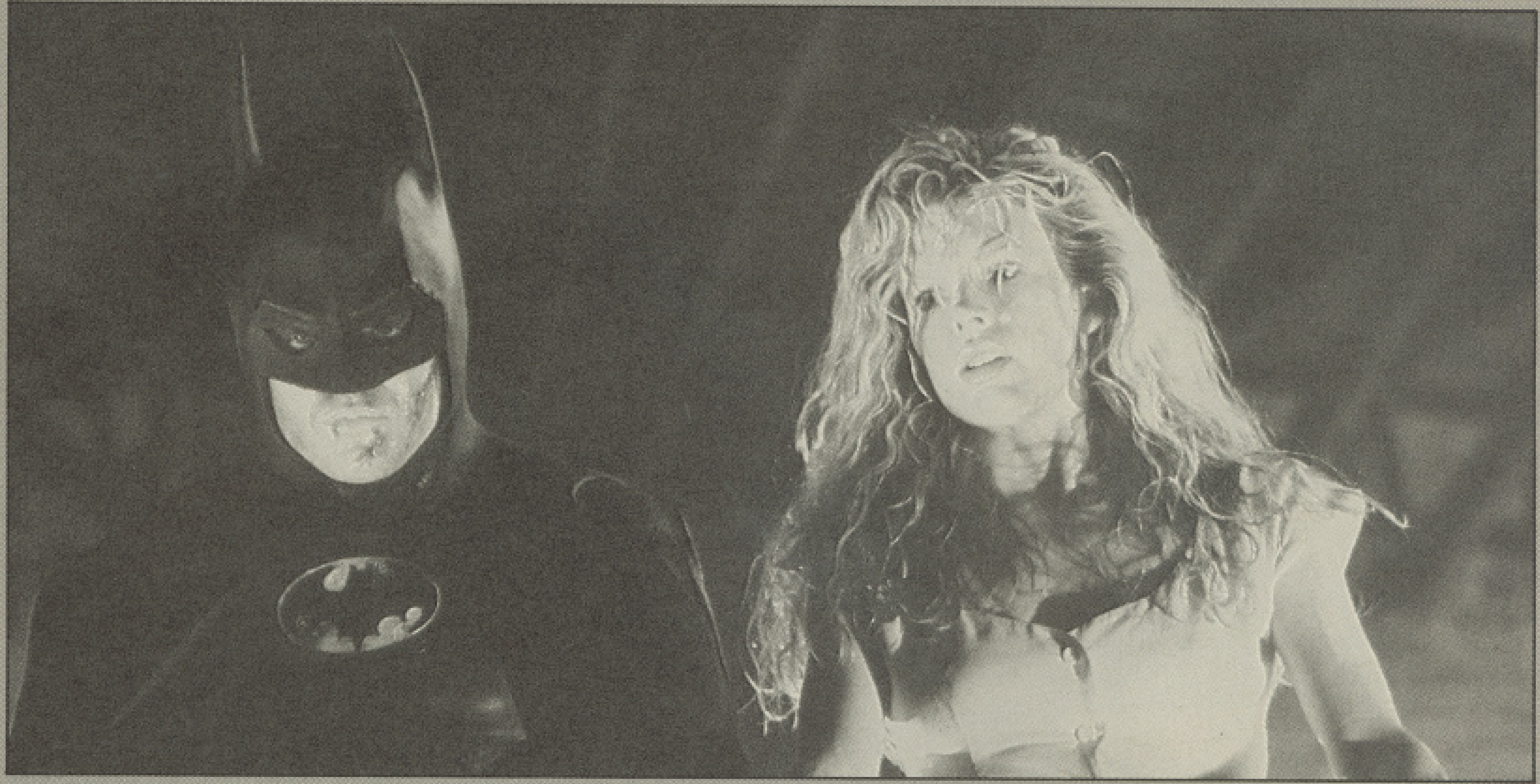




tion from his legions of fans? Perhaps it is that, unlike irradiated or miracle-endowed superheroes from other comics, Batman is one of us – an ordinary man with no super-powers, who spends ten years of his life building himself up to fight crime and avenge the violent deaths of his parents, swearing to stamp out injustice 'to the best of his ability'. The simplicity of the character's origins make him easy to identify with, and more believable in a fantasy universe where an orphan from space wears his underpants outside his tights, and where the mere bite of a spider transforms a man into a superhuman... He is fallible; a haunted, emotional loner constantly at odds with his hidden identity, and the film follows his inner torment and endless bereavement for his lost childhood.

After all, at fifty, it's about time Batman came of age...





The Music

I've seen the future and it will be...' is the first lyric of the first track on the sound track album. And what a fitting and apt lyric to kick off the Batman bandwagon.

Although there are nine individual songs on the album, only four were to be written, and this record only encompasses half of the music in the film – Danny Elfman (noted for his scores in *Beetlejuice* and *Pee Wee's Big Adventure*) provided the 'mood' music throughout. *Prince* doesn't seem to mind though, his presence was requested by *Jack Nicholson* (the Joker), and being signed to the *Warner* record label meant he was the obvious choice anyway. It seems as though *Prince* has made this his album for 1989, putting more effort into each song than you would expect from a normal soundtrack composer.

The songs are well constructed, the album is cunningly arranged, with a mix of several different tastes, and the flavour of each track is unmistakably Prince... With one exception, *Batdance* is probably the weakest track, not being altogether funky nor oozing with Sex – as many Prince songs before it – just settling for plain old dance music. But, as Prince states; I have seen the future and it will be... more *Paisley Park* soundtracks.

The Film

Sceptical Batfans may have slammed the movie before its release, and who wouldn't with the amount of false rumours spread about in the tabloids. But now, upon its release in America, there seems to be little alternative to congratulating the entire team involved.

Michael Keaton is superb as both *Bruce Wayne* and the Dark Knight Detective, and words fail to explain *Jack Nicholson's* performance as the cranky, but always dangerous, *Joker*. Even *Kim Basinger (Vicki Vale)* offers one of her best roles.

On its American premiere 10,000 dedicated Batmaniacs appeared to see the stars parade before them. The lucky few who were able to purchase tickets could rub shoulders with *Sylvester Stallone* (among other stars), and cry and cheer their way through a truly stunning movie.

Once the film has broken all box office records, as it promises to do, and left movie goers, young and old, staggering out of the cinema exhausted, we will have to contend with the fact that it will not be too long before *Batman 2, 3, 4* and many more will come our way to keep our addiction satisfied. Long live Bruce Wayne, long live the Joker, but especially, long live the Batman...

The Game

Ocean are constantly in the habit of biding their time, restricting their output, pooling their resources, and pulling a major licence miracle out of the bag. There can be no better licences to deliver than *Batman – the movie*, and we are to be sure that it is a winner all the way to the bank.

The game is split into four sections, all based upon scenes in the film. The first is based in a Gotham City chemical factory (where the Joker is conceived) which scrolls eight-ways, and is viewed from a side perspective. One feature in this arcade-action adventure is Batman's rope which he can use to swing between levels – something akin to *Bionic Commando*.

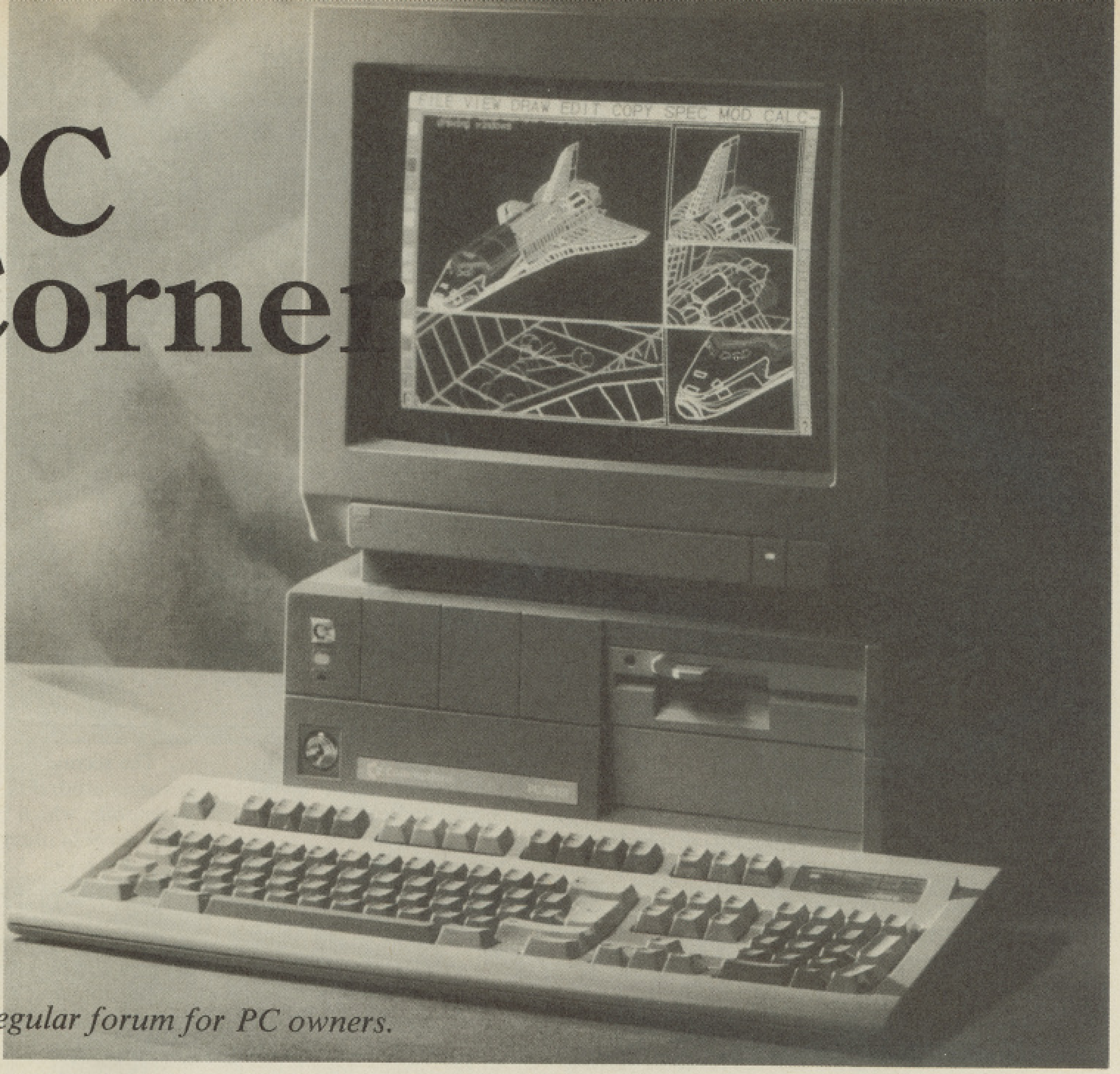
The second subgame revolves around the *Batmobile*, which Batman must drive in order to find the Joker's van. Not only must you avoid other traffic, but you can use your batrope to turn corners at a high velocity.

Section three also features one of Bruce Wayne's modes of transport, this time the *Batwing* is called for. The Joker plans to wipe out Gotham using balloons full of deadly gas. As Batman, you must fly along and cut the wires keeping the balloons affixed to the ground, thus eliminating the danger.

The fourth section is very similar to the first except the Dark Knight detective has to visit a church, for the final confrontation with the Joker.

Ocean predict that the game will be released at the same time as the film in this country – August 11th – so it should prove to be another step in the merchandising boom this autumn and almost certainly the biggest licence in the software industry, ever.

PC Corner



Our regular forum for PC owners.

I have come to rely on my computers a great deal in the course of my day to day life. I have three machines, all of which are made by different manufacturers, all are 'clones' of the IBM PC design and none of them were built by IBM. I use these computers to write software, to do graphics and text for manuals and adverts, control my recording studio, keep up to date with various things and people across three continents and keep my financial records in order for the pleasure of the VAT and tax man. I even sometimes play games on one of them. Before the men in white coats come to take me away I'd like to explain why.

The great strength of the IBM PC family of computers is their flexibility, this is in part due to the availability of a great number of different computers and expansion cards and the sheer volume of software available. The immense number of manufacturers in the PC market means that there is a large spread in the price and power of the hardware. The wide range of expansion cards available means that you can use a PC to perform almost any task by simply adding a relatively simple (and therefore cheap) printed circuit board to the computer. As with the hardware, having lots of software for the machine gives you the advantages that competition within any field will give you – quality and diversity of products.

Another important aspect of the strength of the IBM PC market is that it will have a long life, even IBM with the release of their PS/2 range of computers haven't been able to put a significant damper on the availability and innovation within the PC field. It's nice to know that when you buy a PC that next month or next year you won't be left out on a limb with a computer that has no support or new software.

The speed at which prices are coming down is also amazing, for instance you can now buy a powerful 80386 based AT style PC from *Ness Computing* with 1 megabyte of RAM and a 40 megabyte disk for around £1400 (plus VAT), which is what I paid for my 20 Mbyte XT/PC two years ago. There is also a pocket sized PC being released by Atari soon, which uses memory cards instead of disks and has a serial port for RS232 communications and a lap-top PC from *Yamaha* designed for professional music applications.

Words for the Whys

The IBM PC is basically a 'vanilla' computer, it doesn't have the design compromises that are a feature of a lot of home computers. Home computers have been traditionally designed with a particular goal in mind, this means that they tend to be hot on graphics, or extremely low cost or have some special feature.

The upshot of this is that the basic PC doesn't perform any particular task exceedingly well, but can perform most tasks competently. The success of the PC design is really quite obvious. Just look at the large number of relatively cheap expansion cards on the market; while a lot these are aimed at the business or industrial market there are a few which are aimed specifically at the non-business or 'home' user or can be used for amusement.

Possibly one reason for this success is the mechanical integrity of the PC system. While there is nothing particularly novel about the idea of having additional cards for a personal/home computer, the allowing of space inside the case for them means that the expanded system is robust enough to stand hostile environments (for example those involving pets and small children). The inclusion of the expansion bus in the PC's design also gives us an insight into the antecedents of the PC which in turn explains just why there is so much essential and public domain software available for the PC.

The History Bits

The disk operating system (or DOS) used by the PC is based on one of the earliest systems designed for 8 bit microcomputer systems. This DOS was called CP/M, which stands for Control Program for Microcomputers, the major feature of this software was that it wasn't tied to any one computer. This meant that manufacturers – and even hobbyists – could build a computer for which there was a large amount of common software available. A number of popular PC packages were originally written for CP/M such as Wordstar, dBase II and Super Calc.

The main problem with CP/M was that it was plagued by the fact that there was no standard disk format, the main UK CP/M user group provides disks from its software library in over 120 different disk formats!

The Public Domain

The hobbyist involvement meant that there were a lot of dedicated technical people writing programs for their own use and/or amusement: these people got together in computer clubs in the USA and other parts of the world and released their software in the 'Public Domain' (or PD). These programmers were quite happy to let other people use their software and would even provide limited support, as long as the software wasn't sold commercially. Incidentally, the CP/M operating system isn't dead yet, the Amstrad PCW series uses a version of CP/M version 3.0.

The original IBM PC's operating system was very closely related to an early version of CP/M, which had two results, the first was that people who wanted to upgrade their computer to a 16 bit processor could move to a fairly familiar operating and software environment. The second result was that existing commercial packages could be easily converted (or ported) to the new machine, and since Microsoft, the suppliers of the PC's operating system, also released the DOS software as a separate product, other manufacturers could also build it in to their computers.

The User Groups that distributed the CP/M public domain software now also have MS-DOS programs in their catalogues which range from word-processors, games and CAD systems to the entire King James version of the Bible (6 disks). The sort of software available from the user groups can be very variable in quality, some of it is very good, some obviously not finished. I use a public domain communication pacakge called Procomm, since it is superior to most commercial terminal emulation programs I've seen.

Operating System Enhancements

Mind you, you don't have to put up with the MS-DOS operating system, if you don't want to, there are various ways of 'improving' it. Like the PC hardware, the operating system can be enhanced by adding on what is sometimes called a 'front-end'. Examples of front ends used to improve the user interface of MS-DOS are the graphic environment managers such as *Digital Research's GEM* or *Microsoft Windows* and the menu driven systems such as *Xtree* and *1DIR*.

These programs alter the look and feel of the operating system to make it easier to use, either by implementing the WIMP standard (Windows, Icons, Mouse Pointing device) or by displaying the disk files in your current directory on the screen with a menu of standard commands. *Microsoft Windows* also falls into the category of programs that give your PC the ability to run more than one program at a time – this is called multi-tasking. Other packages that give you this capability are *DESQview* and *Double DOS*.

The End PC

This column has just scratched the surface of the IBM PC compatible computer world. I've tried to give an idea of its roots and thereby explain why this stodgy little computer has become so popular. In future columns I hope to go into greater depth on how to get the PC to do various things, some of which it was not designed to do. I shall also look at some specific programs and expansion cards that I have come across and actually used, not so much as a review, but as a user's comment and as solutions to specific problems which I might have come across.

Organisations & Products Mentioned

This section is for reference, the prices are (where shown) exclusive of VAT and aren't necessarily the cheapest, they're just there as a guide.

Ness AT386-20, PC/AT with 80386, 40 Mbyte Hard Disk, 1 Mbyte RAM
Price – £1395,
Supplier – Ness Computing, 01-739 8410

YAMAHA C1 – LCD Laptop PC/AT, 20 Mbyte Hard Disk, 2 Mbyte RAM
Price – £2995,
Supplier – YAMAHA Pulse, Conduit Street, London

CP/M & MSDOS Users Group, 72 Mill Road, Hawley, Dartford, DA2 7RZ

Compulink User Group, Suite 2, The Sanctuary, Surbiton, KT6 6DU, 01-390 84

Procomm v 2.4.2, communications software
Supplier – Compulink User Group

MicroSoft Windows, Graphics environment for PC
Price – £69,
Supplier – PW Computer Supplies, 01-868 9548

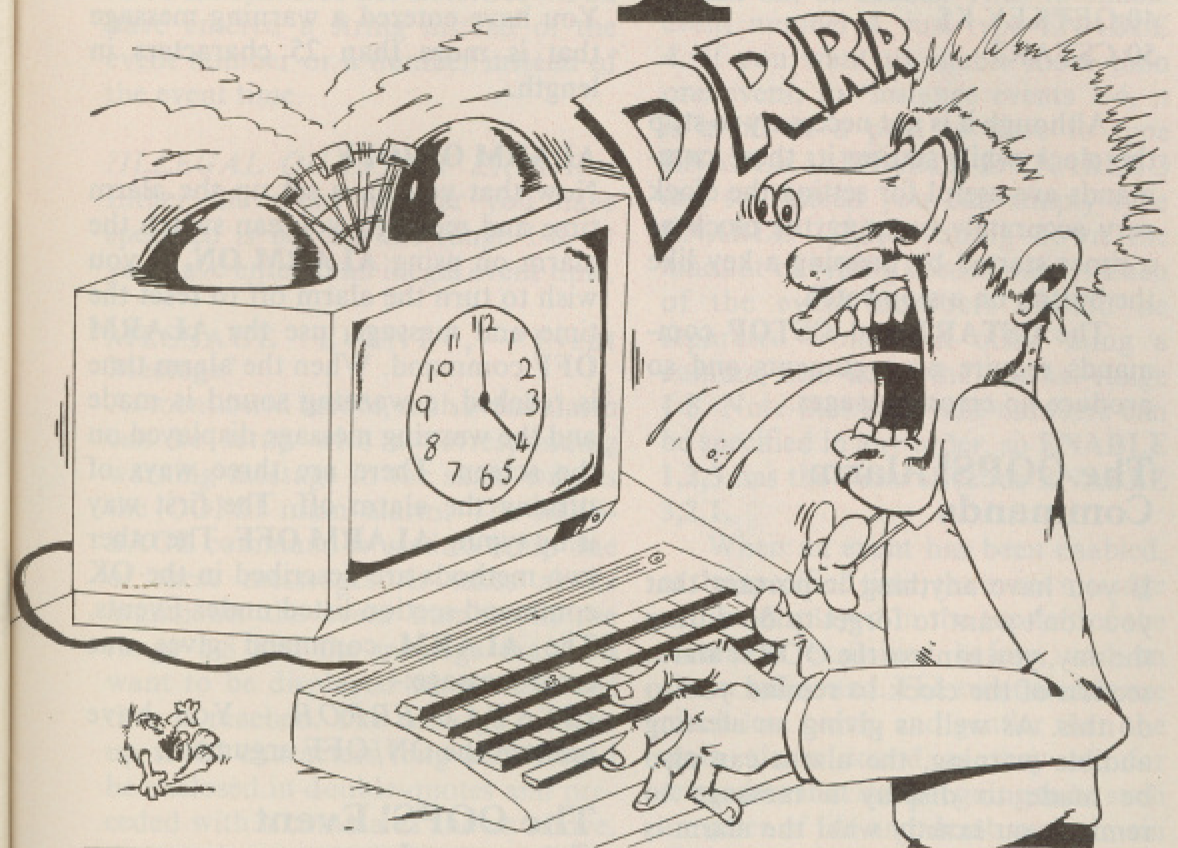
Xtree, DOS file and directory manager from Executive Systems Inc,
Price – £44
Supplier – PW Computer Supplies.

1DIR, DOS and directory manager from Bourbaki Inc,
Supplier – Qubie, 01-871 2855

DESQview, multi-tasking environment
Price – £49
Supplier – CompuAdd, 0800 373535

Double DOS, Soft Logic's multi-tasking system
Price – £39.95
Suppliers – Corporate Software, 07357 5361

Oops!



Make your programming easier with this package of extra commands for the dedicated Plus/4 user

By Mark Everingham

There are two problems with computers. The first is *temporal distortion,* and the second is *human fallibility.* If you are thinking to yourself "What the hell does that mean?" then don't worry – it's really very simple.

Temporal Distortion is the phenomenon by which time seems to become compressed in the prescence of a computer. What this means in practice is that when you pop up to your Plus/4 for "five minutes before tea", you find that tea has long gone and that it's about time you were thinking about some breakfast. The answer to this is, of course, a competent alarm-clock, so OOPS! includes such a clock with advanced user-definable alarms and reminders so that you don't forget lunch or that vital doctor's appointment this afternoon.

The other problem with computers, which I mentioned before, is that although computers, cassette-recorders and disk-drives are relatively reliable (Commodore's are notorious for being very reliable but equally slow), human beings are sadly not so dependable. It is all too easy to crash your Plus/4 or **NEW** the one and only copy of your latest million-seller arcade masterpiece. This event will send most rational human beings into a cry of *OOPS!* (Or similar four letter words).

In order to protect you from these awkward circumstances, OOPS! has an unNEW-type command and a unique automatic program-backup facility making it as difficult as possible to erase or lose an important program.

The features that OOPS! offers are now becoming more and more common as parts of programming languages on the latest generation of 16-bit micros such as the Commodore amiga, but have not until now been available for the BASIC users of 8-bit b home-computers like the Commodore Plus/4.

The OOPS! Basic Commands

OOPS! in practice consists of a program which adds thirty new commands to the normal Commodore Basic Operating-System. This may seem a bit of a daunting prospect at first, but they are all easy to use, and give an idea of how comprehensive the OOPS! system is. The commands can be split into six sections, namely *Clock Commands, Alarm Commands, Event Commands, Saving Commands, Information Commands* and *Miscellaneous other commands.* In a moment I'll demonstrate the use of each of these, but first a short explanation of command syntax.

Each of the OOPS! commands is used in exactly the same way as any standard Commodore Basic commands. Each can be used in Basic Direct mode or in a Basic program, and all can be abbreviated as explained later. Because the commands behave like normal Basic commands, any syntax they have can be replaced with a string variable, for example both the below have exactly the same meaning (to set the 24-hour clock to 23:59:59).

SETCLOCK "23:59:59"

CL$= "23:59:59": SETCLOCK CL$

This property of the commands is useful when writing your own programs, for instance if you can't remember how to work the **SETCLOCK** command, you could write a program which would ask you for the Hours, Minutes and Seconds, and then set the clock for you using something like **SETCLOCK H$+" : "+ M$+" : "+S$** with the hours, minutes and seconds stored in **H$, M$** and **S$** respectively. This facility of the command means that you are not just limited to direct literal commands.

The OOPS! Clock Commands

The OOPS! system includes a 24-Hour clock which does not interfere with the normal Basic TI$ clock and can be made to appear on-screen all the time, whatever else you are doing with your Plus/4 so that you can run programs, print documents, display the disk-directory or whatever you want – the clock will keep running. Below are the commands used to operate the clock.

SETCLOCK "HH:MM:SS"

The **SETCLOCK** command is used to set the initial time of the 24-hour clock, as you will obviously not always start using the OOPS! clock at the same time of the day. It's syntax is the time enclosed within inverted commas. **HH** is the time under the 24-Hour system. **MM** is the number of minutes and **SS** the number of seconds. Each should be separated using a single colon. The command can yield the error messages shown below which can all be trapped using the normal Basic **TRAP** command, and are displayed as normal.

?SYNTAX ERROR – You have omitted the time from the command.

?TYPE MISMATCH ERROR – You have typed a numeric paramater instead of the time string.

?ILLEGAL QUANTITY ERROR – You have entered a time string which is either not of the form "HH:MM:SS" or has a number in it outside the valid range 0-23 for hours or 0-59 for minutes and seconds.

CLOCK ON/OFF
Once you have set the time on the 24-hour clock using the **SETCLOCK** command, you will want to be able to see what time the clock is showing all the time like a normal alarm clock. To do this just type **CLOCK ON**. From then on the clock is always displayed in the top-right corner of the screen. Note that the OOPS! system cannot be used with the high-resolution graphics screen, so the clock is always visible whatever you are doing. However, there may be times when you don't want the clock display to be visible, for example when writing a program that makes use of the full screen. To remove it from the screen type **CLOCK OFF.** Note that although the clock will now be invisible, it is still running and will still display the correct time when a subsequent **CLOCK ON** command is executed. The CLOCK command gives only one error message – SYNTAX ERROR indicating that you have left out, or invalidly replaced the ON/OFF argument.

CSTART and CSTOP
The **CSTART** and **CSTOP** commands are used to respectively start and stop the OOPS! clock. Their main use is for accurate setting of the clock. For example if the time were coming up to 10 o'clock you could use the program below to set the clock very accurately:

```
10 CSTOP
20 SETCLOCK "10:00:00"
30 PRINT "PRESS A KEY WHEN
TIME IS 10 O'CLOCK"
40 GETKEY K$
50 CSTART
```

Although it is not necessary to stop the clock while setting it, these commands are useful for setting the clock very accurately or using the clock as a timer started by pressing a key like the button on a stopwatch.

The **CSTART** and **CSTOP** commands require no arguments and so produce no error messages.

The OOPS! Alarm Commands

If you have anything important that you don't want to forget to do during the day, you can use the OOPS! alarm section of the clock to remind you to do this. As well as giving an alerting audible warning, the alarm can also be made to display a message to remind you exactly what the alarm is set for.

SETALARM "HH:MM"
The **SETALARM** command is used to set up the time at which the alarm will sound. It works in the same way as the **SETCLOCK** command but the alarm time is expressed in only hours and minutes and not also seconds. The error messages the command gives are identical to those for the SETCLOCK command.

WARNING "Alarm Warning Message"
I have mentioned before that as well as making a noise when the alarm time is reached, OOPS! can be made to display a message as a reminder. This message appears inverted and flashing in the top-left corner of the screen to draw attention to itself. The **WARNING** command is used to set up this message for when the alarm goes off. the message should be enclosed in inverted commas and be not more than 25 characters in length. The message may be made up of all alphanumeric characters except control codes such as **RVSON** and **RVSOFF.** The error messsages that the **WARNING** command yields are shown below.

?SYNTAX ERROR – You have left out the warning message.

?TYPE MISMATCH ERROR – You have entered a numeric argument in place of the warning message.

?STRING TOO LONG ERROR – You have entered a warning message that is more than 25 characters in length.

ALARM ON/OFF
Now that you have set up the alarm time and message, you can switch the alarm on using **ALARM ON.** If you wish to turn the alarm off to reset the time and message, use the **ALARM OFF** command. When the alarm time is reached, a warning sound is made and the warning message displayed on the screen. There are three ways of turning the alarm off. The first way is by typing **ALARM OFF.** The other two methods are described in the OK command section listed under Events. The **ALARM** command gives one error message:-
?SYNTAX ERROR – You have omitted the ON/OFF argument.

The OOPS! Event Commands

What are events? Well they are just what they sound like – events, occurences or happenings during the day. Still none the wiser? Well, most of us find that we have more than one thing that we want to remember to do on one day, so a single alarm is of limited use. For this reason, OOPS! has eight 'Event Alarms'. Each of these events works in the same way as the main OOPS! alarm and may be programmed with an event time and associated message. Within a single day you can get your Plus/4 to remind you of up to eight events, enough for those of you with the most densely packed social diaries!

SETEVENT EVENT # , "HH:MM"
The **SETEVENT** command works in the same way as the **SETALARM** command, but is used to set the time at which an event alarm will sound instead of the main OOPS! alarm. Its syntax consists of the number of the event you wish to set, in the range 1-8, followed by the event time enclosed in inverted commas and separated using a comma. The time is entered in an identical format to the **SETA-**

LARM command. The command gives similar error messages, which are shown below:

?SYNTAX ERROR – You have omitted the event number, event time or separating comma.

?TYPE MISMATCH ERROR – You have entered a string instead of the event number or a number instead of the event time.

?ILLEGAL QUANTITY ERROR – Either the event number you have specified is beyond the range 1-8, or you have entered an illegal event time.

MESSAGE EVENT#, "Event Message"
As mentioned before, each event alarm can be setup with a corresponding warning message in the same way as the OOPS! main alarm. The **MESSAGE** command is used to setup one of these messsages. It takes as its arguments the event number in the range 1-8 and the message that you want to be displayed when the event time is reached. The message may be up to 25 characters long and should be enclosed in double-quotes and preceded with a comma as shown above. An event messsage is displayed flashing and inverted in the top-left of the screen in the same manner as the alarm warning set using the **WARNING** command. The **MESSAGE** command gives the error messages that follow:

?SYNTAX ERROR – You have omitted the event number, event message or separating comma.

?TYPE MISMATCH ERROR – You have replaced the event number with a string or the event message with a number.

?ILLEGAL QUANTITY ERROR – The event number you have entered is outside the range 1-8.

?STRING TOO LONG ERROR – The event message you have entered is longer than the allowed 25 characters.

ENABLE EV # 1,EV # 2,...., EV# 8
Once you have setup an event with both event time and event message, you need some way of switching-on or enabling the event, so that when the event time is reached the correct sound is made and message displayed. This is done by means of the **ENABLE** command.

The **ENABLE** command has a flexible syntax in that it can take one or more parameters rather like the Commodore Basic graphic commands. The basic argument is the number of the event that you wish to switch on, or enable. Thus to enable event number 4, just type **ENABLE 4.** If you want to enable more than one event, for instance events 1-4, it is a bit of a pain to have to type **ENABLE 1: ENABLE 2: ENABLE 3** etc. so instead you can simply type **ENABLE 1,2,3,4** cutting down the amount of typing needed vastly. Each of the event numbers should be separated from each other using a comma, and be in the normal range 1-8. Note that the event numbers can be specified in any order, so **ENABLE 1,2,3** has the same effect as **ENABLE 3,2,1.**

When an event has been enabled, at the specified time, the event message will be displayed in the top-left of the screen as explained before, and an alarm will sound. The event alarms are much less harsh, and gentler than the main alarm sound as they are for reminders that are not going to be such a matter of life and death as the main alarm. So if you are of a nervous disposition – use the event alarms instead of the main alarm as they are less of a strain on the heart! The error message that the ENABLE command yields are shown below.

?SYNTAX ERROR – You have either left out all the event numbers, or finished the line with an unneccesary comma.

?TYPE MISMATCH ERROR – You have replaced at least one of the event number with a string instead of a number.

?ILLEGAL QUANTITY ERROR – At least one of the event numbers you have entered is beyond the allowed range 1-8.

DISABLE EV # 1, EV# 2,.....,EV# 8
The **DISABLE** command performs exactly the same function as the **ENABLE** command, but has the effect of switching-off or disabling an event. Its syntax is identical to that of the **ENABLE** command and the error messages it gives have the same meanings as for the **ENABLE** command.

Alarm Priorities and "Shutting Them Up!"

Some of you will have realized while reading the explanations of the OOPS! alarm and events commands that it is perfectly possible to set up all eight events and the main alarm to go off at the same time. What would happen if you did this? After all you can't make 9 noises and display 9 messages all at once, can you? The answer is NO, of course not, so a solution is reached in the form of alarm priorities.

Each event and the main alarm has a priority over the others. The alarm comes first, followed by the eight events in numeric order 1-8. To understand this, imagine that you had setup the alarm and events 1,3,5,7 all to sound at 10 o'clock. Because the main alarm has the highest priority, the alarm noise will be made and the alarm message displayed instead of all the others. "OK" you say, but if I can only hear and see the alarm, then what about events 1,3,5 and 7 – they might be important, too! To find out, read on...

OK (CTRL-0)
When an alarm or event goes off, there are several ways you can turn it off:

(a) If it is the alarm, you can type **ALARM OFF**
(b) If you know which event is occuring you can type **DISABLE EVENT #**
(c) The **OK** command

Methods (a) and (b) are all right, but a bit messy and, if you don't know which event is sounding, then you will have to try each one of the eight in turn to find out. The solution to this is the **OK** command. This command simply switches off whatever is occuring at the moment – the main alarm or the current event. Using this command, you don't have to know which event is occuring, and it also saves you quite a bit of typing.

I discussed earlier the question of what happens when more than one alarm or event is occuring, meaning that you can only see the one with the highest priority – What about the others? Well, when you enter **OK**, it switches-off the event which is currently displayed on screen, and then displays the next one down the priority scale. For instance imagine that the alarm and all the events were set for the same time. When that time was

reached, you would see the alarm message. When you type **OK** return the alarm is switched off and then event 1 is displayed. If you type **OK** return again then event 2 is displayed and so on until all the events have been acknowledged, whereupon the message in the top-left of the screen is erased and the sound turned off again. If you only wanted one alarm, you could use each of the eight events to give you a message of up to 225 characters, using the OK command to read the next section of message, with all events set for the same time.

For those of you who find typing two characters and pressing RETURN a bit arduous, you can achieve the same effect *as typing OK* by pushing CTRL-0 !

OOPS! Saving Commands

The third important section of OOPS! is its automatic saving facility. Using this function, you can get your Plus/4 to automatically save a backup copy of the current program onto disk, so that you don't have to retrace your steps too far if you make a disastrous mistake. You can also make OOPS! mark each file it saves so that you have running log of different versions of your program on disk.

SETSAVING MINUTE
As mentioned before, OOPS! can be made to automatically save backup copies of the program that you are working on, so you need to tells OOPS! when it is to save these backup copies. For instance, if you are writing an important program, you may want it saved every five minutes just to make sure that you don't lose it, but, if you are writing a long program, the saving takes a long time so you may want it to be saved every fifteen minutes or half an hour instead. The **SETSAVING** command is used to specify how often OOPS! should save backup copies. Its only argument is the period of time between successive savings, expressed in minutes. So, if you want your program saved every ten minutes, **SETSAVING 10** will set this up. You can use any time period between every one minute and every hour (60 minutes). The **SETSAVING** error messages are as follows:

?SYNTA ERROR – You have omitted the MINUTE argument.

?TYPE MISMATCH ERROR – You have replaced the **MINUTE** argument with a string.

?ILLEGAL QUANTITY ERROR – The time period you have specified in minutes is beyond the range 1-60.

DEVICE DEV #
For those of you lucky enough to have more than one disk-drive attached to your Plus/4, OOPS! includes the ability to cope with this. When you are writing a program, you might want to have a utility disk containing an Assembler or Font Editor etc. in disk-drive 8, and an OOPS! backup disk in disk-drive 9. To do this, you would simply type **DEVICE 9** – The DEV # argument being the device number of the disk-drive on which OOPS! could save backup files. The device-number can be in the range 8-11 for four disk-drives, but you cannot use device 1 (The Datasette) as this is really too slow for automatic saving. The **DEVICE** error messages are shown below.

?SYNTAX ERROR – You have omitted the device number argument.

?TYPE MISMATCH ERROR – You have replaced the device number with a string instead of a number.

?ILLEGAL DEVICE NUMBER ERROR – The device number you have specified is outside the valid range 8-11.

FILENAME "Filename"
When a backup copy is saved onto disk, it will obviously need to be given a name on the disk, and the **FILENAME** command is used to specify this name. The name argument should be enclosed in inverted commas and may be of up to ten characters in length. It may be made up of any alphanumeric characters but may not include spaces. If you do include a space in the name, the actual name will be truncated by the space character. For instance, entering **FILENAME "HALLO THERE!"** results in a file-name of **"HALLO"**. The **FILENAME** error messages are as follows:

?SYNTAX ERROR – You have failed to enter a file-name.

?TYPE MISMATCH ERROR – You have either replaced the file-name argument with a numeric argument or have not enclosed the name in inverted commas.

?STRING TOO LONG ERROR – The file-name you have entered is longer than ten characters.

OOPS! Backup File Marking

It is a little pointless to keep on saving backup copies of a program onto disk unless you know what each file contains. For example, if you were saving backup copies every two minutes and each of them were called simply "PROGRAM", you might have some files which were saved two hours ago when you were just starting the program, and some which had only just been saved. Obviously you need to know which is which! OOPS! offers several ways of marking your programs. or you can just choose for the last backup copy to be erased every time a new copy is saved. This section discusses the various commands for using the file-marking facility.

VERSION VER #
The first way of marking a backup copy saved by OOPS! is with a version number indicating at what stage of development that backup copy is. You can tell OOPS! which version number of the program you want the first backup copy to be marked with by using the **VERSION** command with **VER #** being the version number. Thus if you want to start with version number 45, enter **VERSION 45.** OOPS! can cope with version numbers in the range 0-999, and this should be quite sufficient unless you are in the habit of saving 10000 backup copies of program! The **VERSION** command gives the following error messages:

?SYNTAX ERROR – You have omitted out the VER # argument.

?TYPE MISMATCH ERROR – You have replaced the version number argument with a string.

?ILLEGAL QUANTITY ERROR – The version number you have specified is outside the valid range 0-9999.

NUMBERMARK
This is the first of the commands to tell OOPS! how to mark backup copies of a program. The **NUMBERMARK** command makes OOPS! add the current version number to the end of the file-name. For example if the file-name were set to "BACKUP" and the version number to 59, the files saved on disk would be under these names:

"BACKUP V50"... "BACKUP V51"... "BACKUP V52"...

As you can see, OOPS! automatically increments the version number each time it saves a backup copy. In the unlikely event that the version number should exceed 9999, it is reset to zero. The **NUMBERMARK** command takes no arguments, so needs no error messages.

TIMEMARK
As an alternative to marking backup copies with a version number, you can add the current clock time to the file-name instead. This allows you to pinpoint the exact version of a program you want, to the nearest minute. For instance, if you had entered **SETSAVING 10: FILENAME "TEST": TIMEMARK,** the backup copies saved would have been the names shown below, assuming that saving started at two o'clock in the afternoon.

"TEST 14:00"... "TEST 14:10"... "TEST 14:20"...

This function of OOPS! is most useful when you are making frequent changes to a program. The **TIMEMARK** command gives no error messages.

UNMARKED
If you don't want the backup copies of your program to be marked with version number or time, you can enter the command **UNMARKED** to disable these facilities. From then on, backup copies will have the name given to them in the **FILENAME** command, with no suffix. However, there is a fallacy in this – You cannot have more than one file with the same name on a disk, so after the first backup copy has been saved, all subsequent saves will not result in a file being saved onto the disk. To overcome this problem, see the next section on the **REPLACING** command.

REPLACING ON/OFF
As more and more backup copies of a program are saved on a disk, so the amount of free space remaining gets smaller and smaller until the disk overflows. If you are working on a long program there may not be room for more than three copies or so on one disk. The answer to this problem and also to saving backups with no suffix is to erase the last backup copy before you save the new one. In this way the amount of room left on a disk will only decrease by the amount you add to a program between saves. For instance if you had entered **VERSION 1: NUMBERMARK: FILENAME "OOPS": REPLACING ON,** the following process would take place:

Save "OOPS" V1"...Erase "OOPS— V1" and Save "OOPS!" V2"...

As you can see, this saves a large amount of space on a disk, but also means that your files are not quite so secure because the last copy has to be erased, and it is also slower than just saving new copies. When the first copy of a program is saved with Replacing On, OOPS! senses that this is the first save, and so does not attempt to erase a nonexistent previous file! **REPLACING** gives only the SYNTAX ERROR message, if you omit ON & OFF.

SAVING ON/OFF
When you have decided what to call your backup copies, how to mark them and whether or not to erase the last one etc. you can enable the OOPS! saving function using SAVING ON. To switch it off again just use SAVING OFF. Once saving is switched on, when it is time to save a backup copy (as dictated using SETSAVING) the following happens:-

1. The line you are entering is erased.
2. The messages below are displayed and their corresponding actions carried out.
ERASING "Last File-name" if REPLACING ON
Last backup copy is erased if REPLACING ON
SAVING "0: New File-name"
3. Your BASIC program is saved.
4. The Computer returns to Direct mode.

As you can see from the above, each time a program is saved the line that you are entering at that time has to be erased. This may seem a bit inconvenient, but when you think about it, the most you can lose is 80 characters of program, whereas if you hadn't saved your program you could have lost all n kilobytes of it!

The saving function of OOPS! only works from Basic Direct mode (When entering programs). If it is time to save a backup copy while a program is running, OOPS! simply waits for the next time and then tries again. In this way it does not interfere with the running of the program.

SAVING gives the SYNTAX ERROR message if the ON & OFF argument is left out.

STORE (CTRL-X)
Sometimes, you may want to save extra backup copies of a program separately from the timed saving function. It is annoying to have to wait ten minutes (or whatever) for the next automatic backup, but you may still want the program to be saved with a version number or the time etc. To save a program at any time, using the marking etc., you can simply type **STORE**. This has the same effect as if it were time to save a backup copy automatically. Additionally, the **STORE** command can be used even when automatic saving has been disabled using **SAVING OFF.**

Again, for those who consider typing five characters a major undertaking, you can achieve the same effect as typing **STORE** by simply pressing CTRL-X . Yes, I would have liked to use CTRL-S too, but Commodore beat us to it, for stopping printing!

OOPS! Information Commands

As you can see, OOPS! has a large number of functions which, when you first use the system, can be more than a little confusing. To help you when you are just starting to use OOPS! I have included in the program five commands to give you information on all aspects of OOPS! from its commands to the current settings of the alarm or the automatic saving. I'll now discuss each of these in turn.

COMMANDS
OOPS! has thirty commands in all, and you may find it hard to remember all of them. If this is the situation, just type **COMMANDS.** This results in a heading and a list of all thirty commands as reference. You should then not find it too hard to remember which command does what.

CLINFO
Apart from the **COMMANDS** command, there are also four specialized information commands which cover the clock, alarm, events and saving functions of OOPS! The first of these is **CLINFO.**

On entering **CLINFO**, a list of information about the OOPS! clock is printed on-screen, which is as follows:-

TIME: HH:MM:SS	– The current time of the clock.
DISPLAY: ON/OFF	– Whether the clock display is on or off as set up by the CLOCK ON/OFF command.
STATUS: STARTED/ STOPPED	– Whether the clock is running or not, as dictated using the CSTART and CSTOP commands.

ALINFO

The **ALINFO** command is similar to the **CLINFO** command, but is used to print information about the OOPS! alarm instead of the clock. It gives the information shown below.

TIME: HH:MM	–The time at which the alarm is set to go off.
STATUS: ON/OFF	– Whether the alarm is enabled or disabled using the ALARM ON/OFF command.
MESSAGE: "Warning"	– The warning message to be displayed when the alarm time is reached, as defined using the WARNING command.

EVINFO

The **EVINFO** command displays a list of information about the current settings for each of OOPS's eight events. All are listed under a set of headings which may be explained as follows:-

	– The event number to which this line of information refers.
TIME	– The time at which this event will occur.
STS	– The status of this event – whether or not it is on or off as defined using the ENABLE and DISABLE commands.
MESSAGE	– The event message assigned to this event using the MESSAGE command.

Each line shows the same information, but for event number as shown on the left of the line.

SVINFO

The last of the OOPS! information commands is the **SVINFO** command which is used to display the current settings for the automatic saving function in OOPS! The list of information the command gives is shown below.

TIMING: MM	– Period between subsequent saves in minutes as defined by the SETSAVING command.
STATUS: ON/OFF	– Whether or not the automatic saving is on or off, as set up using SAVING ON/FF.
MODE: UNMARKED/ NUMBERMARK/ TIMEMARK	– The current method for marking backup copies.
REPLACING: ON/OFF	– Whether or not the replacing or erasing function is enabled by the REPLACING command.
VERSION: VVVV	– The current version number set up using the VERSION command.
DEVICE: DD	– The current saving device-number, as set up using the DEVICE command.
FILENAME: "Filename"	– The current save file-name defined with the FILENAME command.

Miscellaneous Commands

As well as the specific function commands and information commands, I have provided OOPS! with a few extra commands to make life a bit easier when using the system. This section deals with these three remaining commands, namely **WINDOW, RECOVER** and **RESET.**

WINDOW WDW #

You may have noticed that when the clock display is enabled, scrolling the screen causes the clock to blink in an annoying way. You may also have seen that some strange effects can be achieved using the Delete and Insert keys on the top two lines of the screen. It would be better if all work took place in an area of the screen not affected by the clock display or flashing messages, which would remedy the problems outlined above. The answer is the **WINDOW** command.

The **WINDOW** command is used to reduce the usable area of the screen to an area below the clock display so that it does not interfere with any work you're doing. It is the equivalent of moving the cursor to the corners of the area and pressing ESC-T and ESC-8. Two windows are available which are similar to the ESC-N screens, but with the top edge below the clock display. The WINDOW command takes only one argument – **WINDOW 0** enables the ESC-N equivalent window, and **WINDOW 1** the ESC-R equivalent. These are best explained by trying them out yourselves. The edge coordinates for the respective windows are as follows:

WINDOW 0:	Top-left Corner:	(0,2)
	Bottom-Left Corner:	(39,24)
WINDOW 1:	Top-Left Corner:	(1,3)
	Bottom-Left Corner:	(38,23)

The WINDOW command gives the error messages shown below.

?SYNTAX ERROR – You have omitted the window number argument.

?TYPE MISMATCH ERROR – You have mistakenly replaced the window number with a string.

?ILLEGAL QUANTITY ERROR –

The window number you have specified is neither 0 or 1 as it should be.

RECOVER
In addition to the automatic backup facility of OOPS! to stop you losing valuable programs, OOPS! also has an unNEW-type command **RECOVER.** If you enter a program and then erase it from memory using the **NEW** command, you can get it back by typing **RECOVER**. The command will of course not work after you press the RESET button on your Plus/4, doing this erases the OOPS! program code. Some of the time you may be able to get your program back in this situation by reloading the OOPS! program then typing **RECOVER**. You should not try to use the command when you've just switched your computer on, because it will give some funny results with no program data present.

The **RECOVER** command has no arguments so it yields no error messages.

RESET
The final OOPS! command, **RESET**, is used to reset the OOPS! data to its default values. This turns the clock on, resets all the events and saving etc. to the values that they have when you first use OOPS! These default values are setup as typical values such as you might frequently use. These default settings are shown here.

OOPS! Abbreviations

Like the normal Commodore Basic commands, OOPS! commands can be typed abbreviated to save on typing. When typed into a program abbreviated, they are converted into their full form on listing so are just as easy to understand as typing them in fully. A list of the OOPS! command abbreviations is provided here. All abbreviations take exactly the same syntax as the full command.

Command	Arguments	Minimum Abbreviation
SETCLOCK	“HH:MM:SS”	S SHIFT-E
CLOCK	ON/OFF	C SHIFT-L
SETALARM	“HH:MM”	SET SHIFT-A
WARNING	“Message”	W SHIFT-A
ALARM	ON/OFF	A SHIFT-L
SETEVENT	EV , “HH MM”	SET SHIFT-E
MESSAGE	EV , “Message”	M SHIFT-E
ENABLE	EV1,..,EV8	E SHIFT-N
DISABLE	EV1,..,EV8	D SHIFT-I
OK	–	OK
SETSAVING	SAV	SET SHIFT-S
DEVICE	DEV	D SHIFT-E
FILENAME	“Filename”	F SHIFT-I
VERSION	VER	V SHIFT-E
NUMBERMARK	–	N SHIFT-U
TIMEMARK	–	T SHIFT-I
UNMARKED	–	U SHIFT-N
REPLACING	ON/OFF	R SHIFT-E
SAVING	ON/OFF	S SHIFT-A
STORE	–	S SHIFT-T
CSTART	–	C SHIFT-S
CSTOP	–	CST SHIFT-O
RESET	–	RE SHIFT-S
RECOVER	–	RE SHIFT-C
WINDOW	0/1	W SHIFT-I
COMMANDS	–	C SHIFT-O
CLINFO	–	CL SHIFT-I
ALINFO	–	AL SHIFT-I
EVINFO	–	E SHIFT-V
SVINFO	–	S SHIFT-V

OOPS! Kickstart File Generator

You should now have sufficient knowledge to be able to use all of the OOPS! commands competently. I'll now discuss the OOPS! Kickstart File Generator program (Listing # 2).

Most of the time when using OOPS! you will find that you always use the same saving setup, or the same event times and messages, so it is a bit of a waste of time to keep on typing the list of SETEVENT commands or whatever to see these values every time you use OOPS! or every time you reset your Plus/4. It would be much easier to just write a short “Kickstart” program to set up the OOPS! functions so that all you need to do whenever you use OOPS! is to load and RUN this program. This is of course possible, and you can write such a program like any other Basic program using the OOPS! commands, but as an alternative, I have included the OOPS! Kickstart File Generator which can write such a program itself – A program writing another program!

When you run the Kickstart File Generator (Henceforth referred to as KICKGEN), it will ask you a series of questions. When it has done this, it will create a Kickstart program and then save it to disk or cassette. In this section I'll discuss all the questions KICKGEN asks, with the expected replies shown within parentheses.

(1) WINDOW (0/1) Enter which of the windows you wish to start using OOPS1 within: The ESC-N type window 0, or the ESC-R window 1.

(2) CLOCK TIME (HH/MM/SS) Enter the time you wish the OOPS! clock to be set to when you first start using OOPS! Note that when you are entering times into KICKGEN, you mus separate the Hours, Minutes and Seconds with any character EXCEPT a colon. For example 10/00/00, 10-00-00 and 10 00 00 are all valid, but 10:00:00 will not work as the BASIC input routine cannot cope with it.

(3) CLOCK DISPLAY (ON/OFF) Enter whether or not you want the clock display switched ON or OFF when you start using OOPS!

(4) DO YOU WANT TO SET THE ALARM? (YES/NO) If you don't want to bother with setting the OOPS! alarm, just enter NO and go on to section (8). If you do want to set the alarm, follow the steps below.

(5) ALARM TIME (HH:MM) Enter the time you want to set the alarm to.

(6) ALARM WARNING (Message) Enter the message that you want to be displayed on-screen when the alarm time is reached.

(7) ALARM STATUS (ON/OFF) Enter whether or not you want the alarm to be switched ON or OFF when you start using OOPS!

(8) DO YOU WANT TO SET ANY EVENTS? (YES/NO) If you don't want any events set up when you start using OOPS! enter NO and go to step (12). Otherwise, enter YES and follow the steps below.

(9) EVENT X TIME (HH/MM) When you have answered that you do want to set some of the OOPS! events, you will be asked a series of questions for each of the events 1-8. The prompts displayed are of the same form for each event, with X being the event number to which the query refers. E.g. "EVENT 5 TIME" or "EVENT 7 STATUS". The first entry, EVENT TIME is the time you want this particlar event to be set to, but if you don't want to set this event, just press RETURN without entering anything and go on to the next event.

(10) EVENT X MESSAGE (Message) Enter the message you wish to be displayed when the time for event X is reached.

(11) EVENT X STATUS (ON/OFF) Enter whether or not you want event number X to be enabled (ON) or disabled (OFF).

(12) DO YOU WANT TO SET THE SAVING? (YES/NO) If you want to set up the automatic saving facility of OOPS! enter YES and follow the steps below. If not, enter NO and skip to the end of this section.

(13) SAVE TIMING (1-60) Enter the period of time (in minutes) between subsequent automatic saves.

(14) DEVICE NUMBER (8-11) Enter the device-number of the disk-drive on which you wish OOPS! to save its automatic backup copies.

(15) FILENAME (Name) Enter the filename under which you want backup copies of a program to be saved.

(16) VERSION NUMBER (0-9999) Enter the version number that you wish the first backup copy to be marked with when VERSIONMARK is executed.

(17) MARKING MODE (N,T,U) Enter the first character of the mode by which you wish backup copies to be marked. This should be one of (N)umbermark, (T)imemark or (U)nmarked.

(18) REPLACING (ON/OFF) Enter whether or not you wish the previous backup copy to be erased when a new backup copy is saved.

(19) SAVING STATUS (ON/OFF) Enter whether or not you want the automatic saving function of OOPS! to be switched ON or OFF when you first start using the system.

When you have answered all the questions the screen window will clear and the message "COMPILING KICKSTART FILE..." will be displayed. After this, program lines will gradually be displayed which set up all the OOPS! functions as you have specified by answering KICKGEN's questions. When KICKGEN is displaying program lines, what it is doing is actually building a BASIC program into another area of memory from itself. When it has finished this process, KICKGEN displays the message "COMPILING COMPLETE – TAPE/DISK (T/D)?" If you want to save the Kickstart program to Tape, press "T" or if to Disk, press "D". KICKGEN then asks you to insert a disk or tape, and then when you press RETURN it will save the Kickstart program onto the tape or disk..

When KICKGEN has saved the Kickstart program, from that time you can set up your preferred OOPS! functions simply by typing the following.

DLOAD "KICKSTART" if you are using a disk-drive.

LOAD "KICKSTART" if you are using a Datasette as a more secure media.

When the program has loaded, simply type RUN and hit RETURN . Of course, don't forget to load the OOPS! main program before!

That completes our discussion of the OOPS! coomands and KICKGEN program. I'll now discuss the most important aspect of OOPS! – Getting the thing started!

The OOPS! system comes in two parts which are the OOPS! Basic Loader (Listing # 1) and the OOPS! Kickstart File Generator (Listing # 2). The Basic Loader is simply used to create the machine-code OOPS! program, and is used as follows:

Enter Lising # 1 and RUN. The program will give messages to allow you to correct the data lines. Note that you should start the section of data with line 1000. To help you, you can use the line below to set up automatic line numbering, and function key 1 to produce the DATA command with one keypress:-

KEY 1, "DATA" : AUTO 10

When you have entered Listing # 1 and corrected all your mistakes, the program will ask you whether you want to save the OOPS! programs to disk or tape. Press "D" or "T" to choose which and then insert a diskette or tape and press RETURN to save the programs. The programs saved are: (a) BASIC Loader and (b) OOPS! Machine-Code Program. When these programs have been saved, to use the OOPS! system just reset your Plus/4 and type the following:

For Tape Users: **LOAD "OOPS!"** RETURN **RUN** RETURN

For Disk Users: **DLOAD "OOPS!"** RETURN **RUN** RETURN

When you type RUN, the OOPS! commands are enabled, a title screen displayed and the clock switched on. You can then write programs or use OOPS! commands. If you use the RUN/STOP-RESET combination to escape from a machine-code program, typing X RETURN from TEDMON will get you back into Basic and re-enable the OOPS! functions.

As mentioned before, the saving facility of OOPS! cannot be used with a Datasette, but all the clock, alarm and event functions may be used with either disk or tape.

The KICKGEN program (Listing # 2) can be entered just like any other normal BASIC program. It can be saved as shown below.

For Tape Users: **SAVE "OOPS! KICKGEN"** RETURN

For Disk USers: **DSAVE "OOPS! KICKGEN"** RETURN

The OOPS! KICKGEN program may be used either with or without OOPS! resident in memory.

Believe it or not, that concludes this article about OOPS! Now then, I'll just format that disk... there we go... hang on, what does that disk-label say? "OOPS! Master disk"... Oh ****!

```
PROGRAM: LISTING 1
30 REM * OOPS! BASIC LOADER PROGRAM *
60 REM
70 GRAPHIC 1,1:GRAPHIC 0
80 PRINT CHR$(27)"ROOPS! BASIC LOADER PROGRAM"
90 PRINT "WRITTEN IN OCTOBER '88 BY M.EVERINGHAM"
100 PRINT "──────────────────────────────"CHR$(27)"T";
110 TT=0:AD%=4097:FOR LI=1000 TO 5700 STEP 10
120 PRINT "NUMBER OF LINES TO STORE:"(5700-LI)/10"  "
130 CH%=0:FOR BY=0 TO 7:READ DA$
140 DA%=LEN(DA$):IF DA%<2 OR DA%>2 THEN 320
150 DA%=DEC(DA$):POKE AD%+BY,DA%:CH%=CH%+DA%:TT=TT+DA%
160 NEXT BY:READ DA$:IF CH%<>DEC(DA$) THEN 330
170 AD%=AD%+8:NEXT LI:IF TT<>365760 THEN 340
180 PRINT "DATA CORRECT - TAPE OR DISK? (T/D)"
190 DO:GET K$:LOOP UNTIL K$="T"OR K$="D"
200 IF K$="T"THEN POKE 208,1:ELSE POKE 208,8
210 PRINT "INSERT OOPS! ";:IF K$="T"THEN PRINT "TAPE";:ELSE PRINT "DISK";
220 PRINT " AND PRESS RETURN"
230 DO:GET K$:LOOP UNTIL K$=CHR$(13):PRINT "SAVING BASIC LOADER PROGRAM..."
240 SAVE "OOPS! LOADER",PEEK(208)
250 PRINT "SAVING OOPS! SYSTEM PROGRAM..."
260 FOR BY=0 TO 3:POKE 209+BY,PEEK(43+BY):NEXT BY
270 POKE 43,1:POKE 44,16:POKE 45,180:POKE 46,30
280 SAVE "OOPS!",PEEK(208)
290 POKE 43,PEEK(209):POKE 44,PEEK(210):POKE 45,PEEK(211):POKE 46,PEEK(212)
300 PRINT "PROGRAM SAVING COMPLETE"
310 END
320 PRINT "ILLEGAL DATA ITEM FOUND IN LINE"LI:END
330 PRINT "CHECKSUM ERROR FOUND IN LINE"LI:END
340 PRINT "TOTAL CHECKSUM ERROR FOUND:"ABS(365760-TT):END
910 REM
920 REM ************************
930 REM *                      *
940 REM * PROGRAM DATA SECTION *
950 REM *                      *
960 REM * (START AT LINE 1000) *
970 REM *                      *
980 REM ************************
990 REM
1000 DATA 0E,10,00,00,9E,20,34,31,0141
1010 DATA 31,32,3A,A2,00,00,00,4C,018B
1020 DATA 7C,10,A9,63,A0,19,85,22,02F8
1030 DATA 84,23,A0,00,84,0B,88,C8,0326
1040 DATA 20,A5,04,38,F1,22,F0,F7,03FB
1050 DATA C9,80,F0,22,B1,22,30,03,0361
1060 DATA C8,D0,F9,C8,E6,0B,18,98,04FA
1070 DATA 65,22,85,22,90,02,E6,23,02C9
1080 DATA 18,A0,00,B1,22,D0,D9,38,036C
1090 DATA 20,79,04,4C,6A,89,05,0B,01EC
1100 DATA C8,4C,D4,89,AA,A0,63,84,04A2
1110 DATA 22,A0,19,84,23,4C,9E,8B,02F7
1120 DATA C9,80,90,14,C9,9E,B0,10,0414
1130 DATA 38,E9,80,0A,A8,B9,2D,1A,0353
1140 DATA 48,B9,2C,1A,48,4C,73,04,0252
1150 DATA 4C,A1,94,A9,13,A0,10,8D,037A
1160 DATA 0C,03,8C,0D,03,A9,55,A0,0249
1170 DATA 10,8D,0E,03,8C,0F,03,A9,01F5
1180 DATA 61,A0,10,8D,10,03,8C,11,024E
1190 DATA 03,A9,DB,A0,10,8D,02,03,02C9
1200 DATA 8C,03,03,A9,B4,A0,1E,85,0332
1210 DATA 2B,84,2C,A2,00,20,85,15,0237
1220 DATA A9,E8,A0,1D,20,88,90,A2,0428
1230 DATA 88,BD,7B,1B,9D,67,1A,CA,03C3
1240 DATA D0,F7,A2,8C,BD,03,1C,9D,046E
1250 DATA EF,1A,CA,D0,F7,78,A9,76,0531
1260 DATA A0,18,8D,14,03,8C,15,03,0200
1270 DATA 58,60,20,CE,10,4C,12,87,029B
1280 DATA 20,B0,04,38,E9,30,0A,0A,0239
1290 DATA 0A,0A,85,03,C8,20,B0,04,0238
1300 DATA 38,E9,30,05,03,C8,C8,60,0349
1310 DATA 20,79,04,C9,91,D0,06,20,02ED
1320 DATA 73,04,A9,80,60,C9,4F,F0,0408
1330 DATA 03,4C,A1,94,20,73,04,C9,02E4
1340 DATA 46,D0,F6,20,73,04,C9,46,03B2
1350 DATA D0,EF,20,73,04,A9,00,60,035F
1360 DATA 20,84,9D,E0,01,B0,03,4C,0321
1370 DATA 1C,99,E0,09,B0,F9,60,AA,0451
1380 DATA 4A,4A,4A,4A,09,B0,99,46,02C0
1390 DATA 0C,8A,29,0F,09,B0,99,47,0267
1400 DATA 0C,C8,C8,C8,60,20,B0,04,0398
1410 DATA C9,60,90,04,29,DF,D0,02,0397
1420 DATA 29,3F,09,80,60,8E,7B,11,026B
1430 DATA 8C,7C,11,A0,64,8C,01,0C,02B6
1440 DATA 8C,1B,0C,A9,A0,8D,29,0C,02BE
1450 DATA 8D,43,0C,A2,18,98,9D,C2,02CD
1460 DATA 0C,BD,2A,08,09,80,9D,2A,024B
1470 DATA 08,BD,FF,FF,9D,2A,0C,CA,0460
1480 DATA 10,EB,60,C0,09,D0,04,A2,039A
1490 DATA 00,F0,03,98,0A,AA,AD,68,0354
1500 DATA 1A,DD,6C,1A,F0,02,18,60,02E7
1510 DATA AD,69,1A,DD,6D,1A,D0,F6,045A
1520 DATA 8C,7F,1A,C0,09,D0,0A,A2,036A
1530 DATA 98,A0,1A,20,56,11,4C,D0,02F8
1540 DATA 11,B9,8F,1C,18,69,64,AA,0354
1550 DATA A9,1A,69,00,A8,20,56,11,025B
1560 DATA AD,6B,1A,29,28,F0,2F,A2,0344
1570 DATA 84,A0,03,A9,20,D0,0D,AD,037A
1580 DATA 6B,1A,C9,32,90,20,A2,F0,03C2
1590 DATA A0,03,A9,40,8E,0F,FF,8C,03B4
1600 DATA 10,FF,AA,AD,11,FF,29,9F,043E
1610 DATA 09,0F,8D,11,FF,8A,0D,11,025D
1620 DATA FF,8D,11,FF,38,60,A2,FE,04D4
1630 DATA A0,03,8E,0F,FF,8C,10,FF,03DA
1640 DATA 38,60,A5,9A,D0,07,AD,81,03DC
1650 DATA 1A,8D,82,1A,60,C9,C0,F0,041C
1660 DATA F5,A2,08,86,EF,CA,BD,C8,0563
1670 DATA 1C,9D,27,05,CA,10,F7,AD,0363
1680 DATA 80,1A,09,10,8D,80,1A,60,023A
1690 DATA A5,9A,D0,01,60,C9,C0,F0,04E9
1700 DATA FB,A2,05,86,EF,CA,BD,D0,056E
1710 DATA 1C,9D,27,05,CA,10,F7,60,0316
1720 DATA A9,5D,A0,12,20,88,90,A4,0394
1730 DATA CA,88,20,B1,12,A9,63,91,03D2
1740 DATA C8,88,30,05,CC,E7,07,B0,03EF
1750 DATA F6,4C,B1,12,20,49,4E,46,0302
1760 DATA 4F,52,4D,41,54,49,4F,4E,0269
1770 DATA 00,AA,4A,4A,4A,4A,09,30,020E
1780 DATA 20,49,DC,8A,29,0F,09,30,0240
1790 DATA 4C,49,DC,F0,07,A9,8C,A0,043D
1800 DATA 12,4C,88,90,A9,90,A0,12,0361
1810 DATA 4C,88,90,4F,4E,20,00,4F,0270
1820 DATA 46,46,00,A9,3A,4C,49,DC,02E0
1830 DATA 85,03,29,3F,06,03,24,03,0120
1840 DATA 10,02,09,80,70,02,09,40,0156
1850 DATA 4C,49,DC,A9,22,4C,49,DC,03AD
1860 DATA A9,0D,4C,49,DC,A9,20,4C,033C
1870 DATA 49,DC,20,48,9C,C9,08,F0,03EA
1880 DATA 03,4C,1C,99,A0,02,20,B0,0276
1890 DATA 04,C9,3A,F0,03,4C,A1,94,037B
1900 DATA A0,05,20,B0,04,C9,3A,D0,034C
1910 DATA F4,A0,00,20,E1,10,C9,24,0392
1920 DATA B0,DF,48,20,E1,10,C9,60,0411
1930 DATA B0,D7,48,20,E1,10,C9,60,0409
1940 DATA B0,CF,8D,6A,1A,68,8D,69,03EE
1950 DATA 1A,68,8D,68,1A,A9,00,8D,02C7
1960 DATA 6B,1A,60,20,F9,10,F0,09,0307
1970 DATA AD,80,1A,09,80,8D,80,1A,02F7
1980 DATA 60,AD,80,1A,29,7F,8D,80,035C
1990 DATA 1A,A0,09,A9,20,99,10,0C,024E
```

```
2000 DATA 99,45,0C,88,10,F7,60,2
0,02F9
2010 DATA 48,9C,C9,05,F0,03,4C,1
C,030D
2020 DATA 99,A0,02,20,B0,04,C9,3
A,0312
2030 DATA F0,03,4C,A1,94,A0,00,2
0,0334
2040 DATA E1,10,C9,24,B0,E8,48,2
0,03DE
2050 DATA E1,10,C9,60,B0,E0,8D,6
D,04A4
2060 DATA 1A,68,8D,6C,1A,60,A9,9
B,0339
2070 DATA 85,03,A9,1A,85,04,20,4
8,023C
2080 DATA 9C,C9,1A,90,03,4C,4C,C
C,0376
2090 DATA 48,A8,C0,19,F0,09,A9,A
0,040B
2100 DATA 91,03,C8,C0,19,D0,F9,6
8,0466
2110 DATA A8,88,20,46,11,91,03,8
8,02C3
2120 DATA 10,F8,60,20,F9,10,F0,0
9,038A
2130 DATA AD,80,1A,09,40,8D,80,1
A,02B7
2140 DATA 60,AD,80,1A,29,BF,4C,8
E,0369
2150 DATA 13,20,21,11,CA,8A,0A,8
5,0248
2160 DATA 04,20,91,94,20,48,9C,C
9,0316
2170 DATA 05,F0,03,4C,1C,99,A0,0
2,029B
2180 DATA 20,B0,04,C9,3A,F0,03,4
C,0316
2190 DATA A1,94,A0,00,20,E1,10,C
9,03AF
2200 DATA 24,B0,E8,48,20,E1,10,C
9,03DE
2210 DATA 60,B0,E0,A6,04,9D,6F,1
A,03C0
2220 DATA 68,9D,6E,1A,60,20,21,1
1,023F
2230 DATA BD,8F,1C,18,69,B4,85,0
3,0325
2240 DATA A9,1A,69,00,85,04,20,9
1,0266
2250 DATA 94,4C,5F,13,20,21,11,A
D,0251
2260 DATA 7E,1A,1D,97,1C,8D,7E,1
A,028D
2270 DATA 20,79,04,C9,2C,D0,06,2
0,0288
2280 DATA 73,04,4C,ED,13,60,20,2
1,0264
2290 DATA 11,BD,97,1C,49,FF,2D,7
E,0374
2300 DATA 1A,8D,7E,1A,20,79,04,C
9,02A5
2310 DATA 2C,D0,06,20,73,04,4C,0
7,01EC
2320 DATA 14,60,AE,7F,1A,D0,01,6
0,02EC
2330 DATA A0,1C,A9,20,99,00,0C,9
9,02C3
2340 DATA 28,0C,88,D0,F7,AD,11,F
F,0440
2350 DATA 29,9F,8D,11,FF,A9,00,8
D,039B
2360 DATA 7F,1A,E0,09,D0,08,20,9
2,030C
2370 DATA 13,A9,00,85,EF,60,BD,9
7,03E4
2380 DATA 1C,49,FF,2D,7E,1A,8D,7
E,0334
2390 DATA 1A,4C,4A,14,20,84,9D,E
0,02E5
2400 DATA 01,B0,03,4C,1C,99,E0,3
D,02D2
2410 DATA B0,F9,8E,81,1A,8E,82,1
A,03FC
2420 DATA 60,20,84,9D,E0,08,90,0
8,0321
2430 DATA E0,0C,B0,04,8E,83,1A,6
0,032B
2440 DATA A2,09,4C,83,86,20,48,9
C,0304
2450 DATA D0,05,A2,08,4C,83,86,C
9,039D
2460 DATA 0B,90,03,4C,4C,CC,48,A
8,02F2
2470 DATA C0,0A,F0,0A,A9,20,99,8
A,03B0
2480 DATA 1A,C8,C0,0A,D0,F8,68,A
8,0484
2490 DATA 88,20,B0,04,99,8A,1A,8
8,0321
2500 DATA 10,F7,A9,00,8D,86,1A,6
0,033D
2510 DATA 20,E1,9D,A6,14,A4,15,C
0,03D1
2520 DATA 27,90,0B,C0,28,B0,04,E
0,033E
2530 DATA 10,90,03,4C,1C,99,8E,8
4,02B6
2540 DATA 1A,8C,85,1A,60,A9,01,8
5,02D4
2550 DATA 03,AD,80,1A,29,FC,05,0
3,0277
2560 DATA 8D,80,1A,60,A9,02,4C,D
8,0356
2570 DATA 14,A9,00,4C,D8,14,20,F
9,030E
2580 DATA 10,F0,06,A9,53,8D,87,1
A,0330
2590 DATA 60,A9,00,8D,87,1A,60,2
0,02B7
2600 DATA F9,10,F0,0F,AD,81,1A,8
D,03DD
2610 DATA 82,1A,AD,80,1A,09,20,8
D,0299
2620 DATA 80,1A,60,AD,80,1A,29,D
F,0349
2630 DATA 4C,10,15,AD,80,1A,09,0
8,01C9
2640 DATA 8D,80,1A,60,AD,80,1A,2
9,02F7
2650 DATA F7,4C,21,15,A9,FF,A0,0
1,03C2
2660 DATA 91,2B,20,18,88,A5,22,1
8,025B
2670 DATA D8,69,02,85,2D,A5,23,6
9,0326
2680 DATA 00,85,2E,4C,98,8A,A9,A
0,036A
2690 DATA A0,1C,20,88,90,A2,00,A
0,0336
2700 DATA 00,B9,63,19,F0,23,10,1
B,0273
2710 DATA 29,7F,20,49,DC,E8,E0,0
3,03B8
2720 DATA D0,06,A9,0D,A2,00,F0,0
B,0329
2730 DATA BD,E2,1C,18,6D,E7,07,8
5,03B3
2740 DATA CA,D0,03,20,49,DC,C8,D
0,047A
2750 DATA D8,60,20,84,9D,E0,02,9
0,03EB
2760 DATA 03,4C,1C,99,86,03,20,8
8,0235
2770 DATA D8,A5,03,0A,0A,18,69,0
3,0218
2780 DATA A8,A2,03,B9,C0,1C,9D,E
5,0464
2790 DATA 07,88,CA,10,F6,A9,93,4
C,03E7
2800 DATA 49,DC,AD,80,1A,09,10,8
D,0312
2810 DATA 80,1A,AD,87,1A,F0,3F,A
D,03C4
2820 DATA 86,1A,F0,3A,A9,D5,A0,1
C,0404
2830 DATA 20,88,90,A2,00,BD,88,1
A,0339
2840 DATA 20,49,DC,E8,EC,86,1A,D
0,0489
2850 DATA F4,A9,7F,AE,83,1A,A0,0
F,0416
2860 DATA 20,BA,FF,AE,86,1A,E8,8
A,0499
2870 DATA A2,87,A0,1A,20,BD,FF,2
0,03DF
2880 DATA C0,FF,A9,7F,20,C3,FF,A
9,0572
2890 DATA DF,A0,1C,20,88,90,A0,0
0,0373
2900 DATA B9,8A,1A,C9,20,F0,05,C
8,0403
2910 DATA C0,0A,D0,F4,C8,AD,80,1
A,049D
2920 DATA 29,03,C9,01,F0,09,C9,0
2,02BA
2930 DATA F0,5A,C8,98,4C,98,16,8
4,0428
2940 DATA 03,A0,03,AE,84,1A,AD,8
5,0324
2950 DATA 1A,85,62,86,63,A2,90,3
8,0354
2960 DATA 20,CE,A2,20,71,A4,85,2
2,036C
2970 DATA 84,23,EE,84,1A,D0,03,E
E,03F4
2980 DATA 85,1A,AD,85,1A,C9,27,D
0,03AB
2990 DATA 0F,AD,84,1A,C9,10,D0,0
8,030B
3000 DATA A9,00,8D,84,1A,8D,85,1
A,0300
3010 DATA A6,03,A9,56,9D,8A,1A,E
8,03D1
3020 DATA A0,00,20,B0,04,F0,07,9
D,0308
3030 DATA 8A,1A,E8,C8,D0,F4,8A,1
8,04BA
3040 DATA 69,02,D0,33,AD,68,1A,A
A,0347
3050 DATA 4A,4A,4A,4A,09,30,99,8
A,02B4
3060 DATA 1A,8A,29,0F,09,30,99,8
B,0239
3070 DATA 1A,AD,69,1A,AA,4A,4A.4
A,02D2
3080 DATA 4A,09,30,99,8D,1A,8A,2
9,0276
3090 DATA 0F,09,30,99,8E,1A,A9,3
A,026C
3100 DATA 99,8C,1A,98,18,69,07,8
D,02EC
3110 DATA 86,1A,A2,88,A0,1A,20,B
D,0361
3120 DATA FF,AE,83,1A,20,BA,FF,A
6,04C9
3130 DATA 2D,A4,2E,A9,2B,20,D8,F
F,03CA
3140 DATA 20,F8,A8,08,A8,AD,81,1
A,03B8
3150 DATA 8D,82,1A,AD,80,1A,29,E
F,0388
3160 DATA 8D,80,1A,28,90,08,98,A
2,0321
3170 DATA 00,86,EF,4C,7D,A7,A2,0
0,0387
3180 DATA 86,EF,60,A9,E5,A0,1C,2
0,043F
3190 DATA 88,90,20,41,12,A9,EC,A
0,03C0
3200 DATA 1C,20,88,90,AD,68,1A,2
0,02A3
3210 DATA 6A,12,20,94,12,AD,69,1
A,0272
3220 DATA 20,6A,12,20,94,12,AD,6
A,0279
3230 DATA 1A,20,6A,12,A9,F6,A0,1
C,0311
3240 DATA 20,88,90,AD,80,1A,29,8
0,0328
3250 DATA 20,7C,12,A9,01,A0,1D,2
0,0235
3260 DATA 88,90,AD,80,1A,29,08,D
0,0360
3270 DATA 07,A9,0C,A0,1D,4C,88,9
0,02DD
3280 DATA A9,15,A0,1D,4C,88,90,A
9,0388
```

```
3290 DATA 1E,A0,1D,20,88,90,20,41,0274
3300 DATA 12,A9,EC,A0,1C,20,88,90,039B
3310 DATA AD,6C,1A,20,6A,12,20,94,0283
3320 DATA 12,AD,6D,1A,20,6A,12,A9,028B
3330 DATA 01,A0,1D,20,88,90,AD,80,0323
3340 DATA 1A,29,40,20,7C,12,A9,25,01FF
3350 DATA A0,1D,20,88,90,20,AC,12,02D3
3360 DATA A0,00,B9,9B,1A,20,99,12,02D9
3370 DATA C8,C0,19,D0,F5,20,AC,12,0444
3380 DATA 4C,B1,12,A9,30,A0,1D,20,02C5
3390 DATA 88,90,20,41,12,A9,37,A0,030B
3400 DATA 1D,20,88,90,A9,01,85,04,0288
3410 DATA A5,04,A8,09,30,20,49,DC,02CF
3420 DATA A9,2E,20,49,DC,20,B6,12,0304
3430 DATA 98.0A,A8,B9,6C,1A,20,6A,0313
3440 DATA 12,20,94,12,B9,6D,1A,20,0238
3450 DATA 6A,12,20,B6,12,A4,04,AD,02B9
3460 DATA 7E,1A,39,97,1C,20,7C,12,0232
3470 DATA 20,B6,12,A4,04,B9,8F,1C,02F4
3480 DATA A8,A2,19,B9,B4,1A,20,99,03A3
3490 DATA 12,C8,CA,D0,F6,20,B1,12,044D
3500 DATA E6,04,A5,04,C9,09,D0,B0,03E5
3510 DATA 60,A9,62,A0,1D,20,88,90,0360
3520 DATA 20,41,12,A9,6A,A0,1D,20,0263
3530 DATA 88,90,AE,81,1A,A9,00,A0,03AA
3540 DATA 03,20,5F,A4,A9,76,A0,1D,0302
3550 DATA 20,88,90,AD,80,1A,29,20,02C8
3560 DATA 20,7C,12,A9,83,A0,1D,20,02B7
3570 DATA 88,90,AD,80,1A,29,03,85,0310
3580 DATA 03,0A,0A,0A,18,65,03,65,0106
3590 DATA 03,A8,A2,0A,B9,C4,1D,20,0311
3600 DATA 49,DC,C8,CA,D0,F6,A9,90,05B6
3610 DATA A0,1D,20,88,90,AD,87,1A,0343
3620 DATA 20,7C,12,A9,9D,A0,1D,20,02D1
3630 DATA 88,90,AE,84,1A,AD,85,1A,03B0
3640 DATA A0,03,20,5F,A4,A9,AA,A0,03B9
3650 DATA 1D,20,88,90,AE,83,1A,A9,0349
3660 DATA 00,A0,03,20,5F,A4,A9,B7,0326
3670 DATA A0,1D,20,88,90,20,AC,12,02D3
3680 DATA A0,00,B9,8A,1A,C9,20,F0,03D6
3690 DATA 06,20,49,DC,C8,D0,F3,20,03F6
3700 DATA AC,12,4C,B1,12,08,48,8A,02A7
3710 DATA 48,98,48,AD,80,1A,29,08,02A0
3720 DATA F0,49,EE,6B,1A,AD,6B,1A,03DE
3730 DATA C9,64,D0,3F,A9,00,8D,6B,03DD
3740 DATA 1A,F8,AD,6A,1A,18,69,01,02C5
3750 DATA 8D,6A,1A,C9,60.D0,2C,CE,0404
3760 DATA 82,1A,A9,00,8D,6A,1A,AD,0303
3770 DATA 69,1A,18,69,01,8D,69,1A,0215
3780 DATA C9,60,D0,17,A9,00,8D,69,03AF
3790 DATA 1A,AD,68,1A,18,69,01,8D,0258
3800 DATA 68,1A,C9,24,D0,05,A9,00,02ED
3810 DATA 8D,68,1A,D8,AD,80,1A,29,0357
3820 DATA 80,F0,2E,A0,09,A9,64,99,03ED
3830 DATA 1D,0C,88,10,FA,A9,A0,8D,0391
3840 DATA 45,0C,8D,4E,0C,A9,BA,8D,0328
3850 DATA 48,0C,8D,4B,0C,A0,00,AD,02A5
3860 DATA 68,1A,20,30,11,AD,69,1A,0213
3870 DATA 20,30,11,AD,6A,1A,20,30,01E2
3880 DATA 11,AD,80,1A,29,40,F0,07,0288
3890 DATA A0,09,20,84,11,B0,17,A0,02C5
3900 DATA 01,AD,7E,1A,39,97,1C,F0,0322
3910 DATA 05,20,84,11,B0,08,C8,C0,02FA
3920 DATA 09,D0,EE,20,23,14,AD,80,034B
3930 DATA 1A,29,10,D0,12,AD,80,1A,027C
3940 DATA 29,20,F0,0B,AD,82,1A,D0,035D
3950 DATA 06,20,03,12,4C,5A,19,AD,01A7
3960 DATA 43,05,29,04,F0,13,A5,C6,02E3
3970 DATA C9,17,D0,06,20,03,12,4C,0237
3980 DATA 5A,19,C9,26,D0,03,20,29,027E
3990 DATA 12,68,A8,68,AA,68,28,4C,0310
4000 DATA 0E,CE,53,45,54,43,4C,4F,02A6
4010 DATA 43,CB,43,4C,4F,43,CB,53,034D
4020 DATA 45,54,41,4C,41,52,CD,57,02DD
4030 DATA 41,52,4E,49,4E,C7,41,4C,02CC
4040 DATA 41,52,CD,53,45,54,45,56,02E7
4050 DATA 45,4E,D4,4D,45,53,53,41,02E0
4060 DATA 47,C5,45,4E,41,42,4C,C5,0333
4070 DATA 44,49,53,41,42,4C,C5,4F,02C3
4080 DATA CB,53,45,54,53,41,56,49,02EA
4090 DATA 4E,C7,44,45,56,49,43,C5,0345
4100 DATA 46,49,4C,45,4E,41,4D,C5,02C1
4110 DATA 56,45,52,53,49,4F,CE,4E,02F4
4120 DATA 55,4D,42,45,52,4D,41,52,025B
4130 DATA CB,54,49,4D,45,4D,41,52,02DA
4140 DATA CB,55,4E,4D,41,52,4B,45,02DE
4150 DATA C4,52,45,50,4C,41,43,49,02C4
4160 DATA 4E,C7,53,41,56,49,4E,C7,035D
4170 DATA 53,54,4F,52,C5,43,53,54,02F7
4180 DATA 41,52,D4,43,53,54,4F,D0,0370
4190 DATA 52,45,53,45,D4,52,45,43,02DD
4200 DATA 4F,56,45,D2,57,49,4E,44,02EE
4210 DATA 4F,D7,43,4F,4D,4D,41,4E,02E1
4220 DATA 44,D3,43,4C,49,4E,46,CF,0352
4230 DATA 41,4C,49,4E,46,CF,45,56,02D4
4240 DATA 49,4E,46,CF,53,56,49,4E,02EC
4250 DATA 46,CF,00,BA,12,03,13,27,021E
4260 DATA 13,56,13,83,13,99,13,D5,0293
4270 DATA 13,EC,13,06,14,22,14,5C,01BE
4280 DATA 14,71,14,85,14,B8,14,D5,02D3
4290 DATA 14,E4,14,E9,14,EE,14,FF,040A
4300 DATA 14,A2,15,1B,15,24,15,AB,01DF
4310 DATA 10,2C,15,7A,15,46,15,D3,020E
4320 DATA 16,27,17,73,17,D9,17,48,0216
4330 DATA 4D,53,54,48,4D,48,4D,48,0266
4340 DATA 4D,48,4D,48,4D,48,4D,48,0254
4350 DATA 4D,48,4D,48,4D,00,00,00,0177
4360 DATA 00,00,00,00,00,00,31,32,0063
4370 DATA 3A,31,32,33,34,35,36,37,01A6
4380 DATA 38,39,30,31,32,33,34,35,01A0
4390 DATA 36,00,31,32,33,34,35,36,016B
4400 DATA 37,38,39,30,31,32,33,34,01A2
4410 DATA 35,36,37,38,39,30,31,32,01A6
4420 DATA 33,34,35,31,32,33,34,35,019B
4430 DATA 36,37,38,39,30,31,32,33,01A4
4440 DATA 34,35,36,37,38,39,30,31,01AB
4450 DATA 32,33,34,35,31,32,33,34,0198
4460 DATA 35,36,37,38,39,30,31,32,01A6
4470 DATA 33,34,35,36,37,38,39,30,01AA
4480 DATA 31,32,33,34,35,31,32,33,0195
4490 DATA 34,35,36,37,38,39,30,31,01AB
4500 DATA 32,33,34,35,36,37,38,39,01AC
4510 DATA 30,31,32,33,34,35,31,32,0192
4520 DATA 33,34,35,36,37,38,39,30,01AA
4530 DATA 31,32,33,34,35,36,37,38,01A4
4540 DATA 39,30,31,32,33,34,35,31,0199
4550 DATA 32,33,34,35,36,37,38,39,01AC
4560 DATA 30,31,32,33,34,35,36,37,019C
4570 DATA 38,39,30,31,32,33,34,35,01A0
```

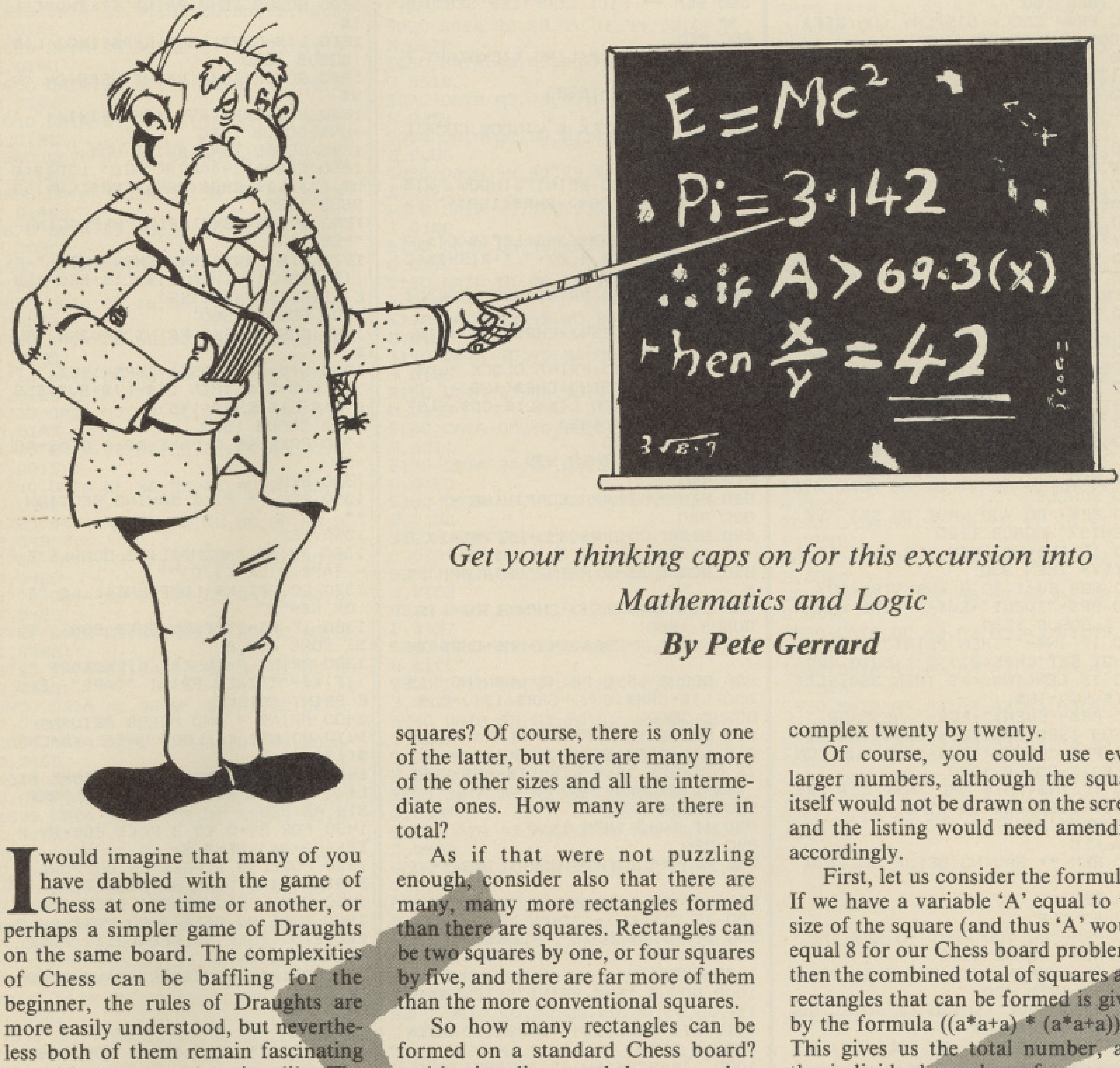
```
4580 DATA 31,32,33,34,35,36,37,3
8,01A4
4590 DATA 39,30,31,32,33,34,35,3
6,019E
4600 DATA 37,38,39,30,31,32,33,3
4,01A2
4610 DATA 35,31,32,33,34,35,36,3
7,01A1
4620 DATA 38,39,30,31,32,33,34,3
5,01A0
4630 DATA 36,37,38,39,30,31,32,3
3,01A4
4640 DATA 34,35,31,32,33,34,35,3
6,019E
4650 DATA 37,38,39,30,31,32,33,3
4,01A2
4660 DATA 35,36,37,38,39,30,31,3
2,01A6
4670 DATA 33,34,35,00,00,00,00,0
0,009C
4680 DATA 00,07,30,08,45,11,00,1
2,00A7
4690 DATA 30,15,00,17,00,19,00,0
0,0075
4700 DATA 00,00,00,88,0A,00,08,0
1,009B
4710 DATA 00,00,00,30,3A,50,52,4
F,015B
4720 DATA 47,52,41,4D,20,20,20,2
0,01A7
4730 DATA 2E,2E,2E,2E,2E,00,A0,8
F,0215
4740 DATA 8F,90,93,A1,A0,82,99,A
0,04AE
4750 DATA 8D,AE,92,A0,85,96,85,9
2,049F
4760 DATA 89,8E,87,88,81,8D,A0,9
4,0468
4770 DATA 89,8D,85,A0,94,8F,A0,8
7,0485
4780 DATA 85,94,A0,95,90,AC,A0,8
C,04B6
4790 DATA 81,9A,99,82,8F,8E,85,9
3,046B
4800 DATA A0,99,8F,95,A7,8C,8C,A
0,04BC
4810 DATA 82,85,A0,8C,81,94,85,A
0,046D
4820 DATA 86,8F,92,A0,97,8F,92,8
B,048A
4830 DATA A0,A0,94,89,8D,85,A0,8
6,0495
4840 DATA 8F,92,A0,93,8F,8D,85,A
0,0495
4850 DATA 85,8C,85,96,85,8E,93,8
5,0457
4860 DATA 93,A0,89,94,A7,93,A0,8
A,04B4
4870 DATA 95,93,94,A0,81,82,8F,9
5,0483
4080 DATA 94,A0,8C,95,8E,83,88,9
4,0482
4890 DATA 89,8D,85,88,8F,97,A0,8
1,046A
4900 DATA 82,8F,95,94,A0,81,A0,8
3,047E
4910 DATA 8F,86,86,85,85,A0,82,9
2,0459
4920 DATA 85,81,8B,BF,99,8F,95,A
0,04AD
4930 DATA 83,81,8E,A0,93,94,8F,9
0,0478
4940 DATA A0,97,8F,92,8B,89,8E,8
7,0481
4950 DATA A0,8E,8F,97,A1,97,8F,9
5,04B0
4960 DATA 8C,84,8E,A7,94,A0,99,8
F,04A1
4970 DATA 95,A0,8C,89,8B,85,A0,8
4,047E
4980 DATA 89,8E,8E,85,92,BF,89,A
7,04AB
4990 DATA 84,A0,93,81,99,A0,89,9
4,048E
5000 DATA A7,93,A0,94,89,8D,85,A
0,04A9
5010 DATA 86,8F,92,A0,82,85,84,0
0,03D2
5020 DATA 19,32,4B,64,7D,96,AF,0
1,02BD
5030 DATA 02,04,08,10,20,40,80,0
D,010B
5040 DATA 4F,4F,50,53,21,20,43,4
F,0214
5050 DATA 4D,4D,41,4E,44,53,0D,A
3,0270
5060 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5070 DATA A3,A3,A3,A3,A3,0D,00,1
8,0354
5080 DATA 02,00,27,17,03,01,26,1
B,0085
5090 DATA 44,53,54,4F,52,45,0D,1
B,01F9
5100 DATA 44,4F,4B,0D,0D,45,52,4
1,01D0
5110 DATA 53,49,4E,47,20,00,1B,5
0,01BC
5120 DATA 91,00,0D,1A,0D,43,4C,4
F,01A3
5130 DATA 43,4B,00,20,20,20,54,4
9,018B
5140 DATA 4D,45,3A,20,00,0D,44,4
9,0186
5150 DATA 53,50,4C,41,59,3A,20,0
0,01E3
5160 DATA 0D,20,53,54,41,54,55,5
3,0211
5170 DATA 3A,20,00,53,54,4F,50,5
0,01F0
5180 DATA 45,44,0D,00,53,54,41,5
2,01D0
5190 DATA 54,45,44,0D,00,0D,41,4
C,0184
5200 DATA 41,52,4D,00,0D,4D,45,5
3,01D2
5210 DATA 53,41,47,45,3A,20,00,0
D,0187
5220 DATA 45,56,45,4E,54,00,23,2
0,01C5
5230 DATA 20,54,49,4D,45,20,20,5
3,01E2
5240 DATA 54,53,20,4D,45,53,53,4
1,0240
5250 DATA 47,45,0D,A3,20,20,A3,A
3,02C2
5260 DATA A3,A3,20,20,A3,A3,A3,2
0,038F
5270 DATA A3,A3,A3,A3,A3,A3,A3,0
D,0482
5280 DATA 00,0D,53,41,56,49,4E,4
7,01D5
5290 DATA 00,20,20,20,54,49,4D,4
9,0193
5300 DATA 4E,47,3A,20,00,0D,20,2
0,013C
5310 DATA 20,53,54,41,54,55,53,3
A,023E
5320 DATA 20,00,0D,20,20,20,20,2
0,00CD
5330 DATA 4D,4F,44,45,3A,20,00,0
D,018C
5340 DATA 52,45,50,4C,41,43,49,4
E,024E
5350 DATA 47,3A,20,00,0D,20,20,5
6,0144
5360 DATA 45,52,53,49,4F,4E,3A,2
0,022A
5370 DATA 00,0D,20,20,20,44,45,5
6,014C
5380 DATA 49,43,45,3A,20,00,0D,2
0,0158
5390 DATA 46,49,4C,45,4E,41,4D,4
5,0241
5400 DATA 3A,20,00,55,4E,4D,41,5
2,01DD
5410 DATA 4B,45,44,20,20,4E,55,4
D,0204
5420 DATA 42,45,52,4D,41,52,4B,5
4,0258
5430 DATA 49,4D,45,4D,41,52,4B,2
0,0226
5440 DATA 20,20,4D,49,4E,53,00,1
1,0188
5450 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5460 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5470 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5480 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5490 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5500 DATA 20,12,AC,A2,BB,A2,A2,A
C,042B
5510 DATA A2,BB,A2,A2,AC,92,A1,2
0,04A0
5520 DATA 20,57,52,49,54,54,45,4
E,024D
5530 DATA 20,49,4E,20,4F,43,54,4
F,020C
5540 DATA 42,45,52,20,31,39,38,3
8,01D3
5550 DATA 0D,20,92,A1,12,20,A1,A
1,02D4
5560 DATA 92,A1,A1,12,A2,A1,92,B
C,0477
5570 DATA 12,A2,92,A1,A1,0D,20,9
2,0347
5580 DATA A1,12,A2,A1,92,BC,BE,A
1,04A3
5590 DATA 12,20,20,A2,92,BE,12,A
C,0302
5600 DATA 92,A1,20,20,20,20,42,5
9,024E
5610 DATA 20,4D,41,52,4B,20,45,5
6,0206
5620 DATA 45,52,49,4E,47,48,41,4
D,024B
5630 DATA 0D,20,12,A2,A2,A2,A2,A
2,0369
5640 DATA A2,A2,A2,A2,A2,A2,92,B
E,051C
5650 DATA 0D,A3,A3,A3,A3,A3,A3,A
3,0482
5660 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5670 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5680 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5690 DATA A3,A3,A3,A3,A3,A3,A3,A
3,0518
5700 DATA A3,00,00,00,00,00,00,0
0.00A3
```

PROGRAM: LISTING 2

```
10 REM **************************
*******
20 REM *
     *
30 REM * OOPS! KICKSTART FILE CR
EATOR *
40 REM *
     *
50 REM **************************
*******
60 REM
70 PRINTCHR$(27)"ROOPS! KICKSTAR
T FILE GENERATOR"
80 PRINT"WRITTEN IN OCTOBER '88
 BY M.EVERINGHAM"
90 PRINT"
                  "CHR$(27)"T";
100 Y$="YES":N$="NO":O1$="OFF":O
2$="ON"
110 DIM ET$(8),EM$(8),ES$(8)
120 REM
130 REM ** CLOCK & WINDOW SETUP
**
```

```
140 REM
150 PR$="WINDOW (0/1):":GOSUB 15
20:WI$=IN$:IF WI$<"0" OR WI$>"1"
THEN 150
160 PR$="CLOCK TIME (HH/MM/SS):"
:GOSUB 1520:CT$=IN$:IF LEN(CT$)<
>8 THEN 160
170 PR$="CLOCK DISPLAY (ON/OFF):
":GOSUB 1520:CD$=IN$
180 IF CD$<>O1$ AND CD$<>O2$ THE
N 170
190 REM
200 REM ** ALARM SETUP SECTION *
*
210 REM
220 SCNCLR
230 PR$="DO YOU WANT TO SET THE
ALARM?":GOSUB 1520
240 IF IN$=N$ THEN 320:ELSE IF I
N$<>Y$ THEN 230
250 PR$="ALARM TIME (HH/MM):":GO
SUB 1520:AT$=IN$:IF LEN(AT$)<>5
THEN 250
260 PR$="ALARM WARNING:":GOSUB 1
520:AM$=LEFT$(IN$,25)
270 PR$="ALARM STATUS (ON/OFF):"
:GOSUB 1520:AS$=IN$
280 IF AS$<>O1$ AND AS$<>O2$ THE
N 270
290 REM
300 REM ** EVENTS SETUP SECTION
**
310 REM
320 SCNCLR
330 PR$="DO YOU WANT TO SET ANY
EVENTS?":GOSUB 1520
340 IF IN$=N$ THEN 460:ELSE IF I
N$<>Y$ THEN 330
350 FOR EV=1 TO 8:EV$=STR$(EV)
360 PR$="EVENT"+EV$+" TIME (HH/M
M):":GOSUB 1520
370 IF IN$=""THEN PRINT"
NOT SET"CHR$(27)"Q":GOTO 420
380 IF LEN(IN$)<>5 THEN 360:ELSE
ET$(EV)=IN$
390 PR$="EVENT"+EV$+" MESSAGE:":
GOSUB 1520:EM$(EV)=LEFT$(IN$,25)
400 PR$="EVENT"+EV$+" STATUS (ON
/OFF):":GOSUB 1520
410 IF IN$<>O1$ AND IN$<>O2$ THE
N 400:ELSE ES$(EV)=IN$
420 NEXT EV
430 REM
440 REM ** SAVING SETUP SECTION
**
450 REM
460 SCNCLR
470 PR$="DO YOU WANT TO SET THE
SAVING?":GOSUB 1520
480 IF IN$=N$ THEN 670:ELSE IF I
N$<>Y$ THEN 470
490 PR$="SAVE TIMING (1-60):":GO
SUB 1520:TS$=IN$
500 IF VAL(TS$)<1 OR VAL(TS$)>60
 THEN 490
510 PR$="DEVICE NUMBER (8-11):":
GOSUB 1520:DE$=IN$
520 IF VAL(DE$)<8 OR VAL(DE$)>11
 THEN 510
530 PR$="FILENAME:":GOSUB 1520:F
L$=LEFT$(IN$,10)
540 PR$="VERSION NUMBER (0-9999)
:":GOSUB 1520:UN$=IN$
550 IF VAL(UN$)<0 OR VAL(UN$)>99
99 THEN 540
560 PR$="MARKING MODE (N,T,U):":
GOSUB 1520:IF INSTR("NTU",IN$)=0
 THEN 560
570 IF IN$="U"THEN MM$="UNMARKED
"
580 IF IN$="N"THEN MM$="NUMBERMA
RK"
590 IF IN$="T"THEN MM$="TIMEMARK
"
600 PRINT"":"CHR$(27)"K"MM$""
610 PR$="REPLACING (ON/OFF):":GO
SUB 1520:RE$=IN$:IF RE$<>O1$ AND
 RE$<>O2$  THEN 610
620 PR$="SAVING STATUS (ON/OFF):
":GOSUB 1520:SV$=IN$
630 IF SV$<>O1$ AND SV$<>O2$ THE
N 620
640 REM
650 REM ** FILE COMPILER SECTION
 **
660 REM
670 PRINT"COMPILING KICKSTART F
ILE..."
680 LI%=10:AD=16385
690 REM
700 REM ** CLOCK & WINDOW COMPIL
ING **
710 REM
720 GOSUB 1640:PRINT"WINDOW "WI$
730 LI$=CHR$(254)+CHR$(152)+" "+
WI$:GOSUB 1560
740 LI$=" "+CHR$(34)+LEFT$(CT$,2
)+":"+MID$(CT$,4,2)+":"+RIGHT$(C
T$,2)+CHR$(34)
750 GOSUB 1640:PRINT"SETCLOCK"LI
$
760 LI$=CHR$(254)+CHR$(128)+LI$:
GOSUB 1560
770 GOSUB 1640:PRINT"CLOCK "CD$
780 LI$=CHR$(254)+CHR$(129)+" ":
IF CD$=O1$ THEN LI$=LI$+CD$:ELSE
 LI$=LI$+CHR$(145)
790 GOSUB 1560
800 IF AT$=""THEN 930
810 REM
820 REM ** ALARM COMPILING **
830 REM
840 LI$=" "+CHR$(34)+LEFT$(AT$,2
)+":"+RIGHT$(AT$,2)+CHR$(34)
850 GOSUB 1640:PRINT"SETALARM"LI
$
860 LI$=CHR$(254)+CHR$(130)+LI$:
GOSUB 1560
870 LI$=" "+CHR$(34)+AM$+CHR$(34
)
880 GOSUB 1640:PRINT"WARNING"LI$
890 LI$=CHR$(254)+CHR$(131)+LI$:
GOSUB 1560
900 GOSUB 1640:PRINT"ALARM "AS$
910 LI$=CHR$(254)+CHR$(132)+" ":
IF AS$=O1$ THEN LI$=LI$+AS$:ELSE
 LI$=LI$+CHR$(145)
920 GOSUB 1560
930 IF EV=0 THEN 1110
940 REM
950 REM ** EVENTS COMPILING **
960 REM
970 FOR EV=1 TO 8
980 IF ET$(EV)=""THEN 1050
990 LI$=STR$(EV)+","+CHR$(34)+LE
FT$(ET$(EV),2)+":"+RIGHT$(ET$(EV
),2)+CHR$(34)
1000 GOSUB 1640:PRINT"SETEVENT"L
I$
1010 LI$=CHR$(254)+CHR$(133)+LI$
:GOSUB 1560
1020 LI$=STR$(EV)+","+CHR$(34)+E
M$(EV)+CHR$(34)
1030 GOSUB 1640:PRINT"MESSAGE"LI
$
1040 LI$=CHR$(254)+CHR$(134)+LI$
:GOSUB 1560
1050 NEXT EV
1060 LI$=" ":FOR EV=1 TO 8
1070 IF ES$(EV)=O2$ THEN LI$=LI$
+CHR$(48+EV)+","
1080 NEXT:LI$=LEFT$(LI$,LEN(LI$)
-1):IF LI$=""THEN 1110
1090 GOSUB 1640:PRINT"ENABLE"LI$
1100 LI$=CHR$(254)+CHR$(135)+LI$
:GOSUB 1560
1110 IF TS$=""THEN 1320
1120 REM
1130 REM ** SAVING COMPILING **
1140 REM
1150 GOSUB 1640:PRINT"SETSAVING
"TS$
1160 LI$=CHR$(254)+CHR$(138)+" "
+TS$:GOSUB 1560
1170 GOSUB 1640:PRINT"DEVICE "DE
$
1180 LI$=CHR$(254)+CHR$(139)+" "
+DE$:GOSUB 1560
1190 LI$=" "+CHR$(34)+FL$+CHR$(3
4)
1200 GOSUB 1640:PRINT"FILENAME"L
I$
1210 LI$=CHR$(254)+CHR$(140)+LI$
:GOSUB 1560
1220 GOSUB 1640:PRINT"VERSION "V
N$
1230 LI$=CHR$(254)+CHR$(141)+" "
+VN$:GOSUB 1560
1240 GOSUB 1640:PRINT MM$
1250 CN%=141+INSTR("NTU",LEFT$(M
M$,1)):LI$=CHR$(254)+CHR$(CN%):G
OSUB 1560
1260 GOSUB 1640:PRINT"REPLACING
"RE$
1270 LI$=CHR$(254)+CHR$(145)+" "
:IF RE$=O1$ THEN LI$=LI$+RE$:ELS
E LI$=LI$+CHR$(145)
1280 GOSUB 1560
1290 GOSUB 1640:PRINT"SAVING "SV
$
1300 LI$=CHR$(254)+CHR$(146)+" "
:IF SV$=O1$ THEN LI$=LI$+SV$:ELS
E LI$=LI$+CHR$(145)
1310 GOSUB 1560
1320 POKE AD,0:POKE AD+1,0:AD=AD
+2
1330 REM
1340 REM ** FILE SAVING SECTION
**
1350 REM
1360 PRINT "COMPILING COMPLETE
- TAPE/DISK (T/D)?"
1370 DO:GET K$:LOOP UNTIL K$="T"
 OR K$="D"
1380 IF K$="T"THEN POKE 208,1:EL
SE POKE 208,8
1390 PRINT "INSERT KICKSTART ";
:IF K$="T"THEN PRINT "TAPE";:ELS
E PRINT "DISK";
1400 PRINT " AND PRESS RETURN"
1410 DO:GET K$:LOOP UNTIL K$=CHR
$(13)
1420 PRINT "SAVING KICKSTART FI
LE...":POKE 213,AD AND 255:POKE
214,AD/256
1430 FOR BY=0 TO 3:POKE 209+BY,P
EEK(43+BY):NEXT BY
1440 POKE 43,1:POKE 44,64:POKE 4
5,PEEK(213):POKE 46,PEEK(214)
1450 SAVE "KICKSTART",PEEK(208)
1460 POKE 43,PEEK(209):POKE 44,P
EEK(210):POKE 45,PEEK(211):POKE
46,PEEK(212)
1470 PRINT"KICKSTART SAVING COM
PLETE"
1480 END
1490 REM
1500 REM ** USER INPUT ROUTINE *
*
1510 REM
1520 IN$="":PRINT PR$;CHR$(27);:
INPUT IN$:PRINT:RETURN
1530 REM
1540 REM ** COMPILE LINE ROUTINE
 **
1550 REM
1560 LE%=LEN(LI$):NA=AD+LE%+5
1570 POKE AD,NA AND 255:POKE AD+
1,NA/256
1580 POKE AD+2,LI% AND 255:POKE
AD+3,LI%/256
1590 FOR CH=1 TO LE%:POKE AD+3+C
H,ASC(MID$(LI$,CH,1)):NEXT CH:PO
KE AD+3+CH,0
1600 AD=NA:LI%=LI%+10:RETURN
1610 REM
1620 REM ** LINE NUMBER ROUTINE
**
1630 REM
1640 PRINTUSING"###";LI%;:PRINT"
";:RETURN
```

Logically Speaking



Get your thinking caps on for this excursion into Mathematics and Logic

By Pete Gerrard

I would imagine that many of you have dabbled with the game of Chess at one time or another, or perhaps a simpler game of Draughts on the same board. The complexities of Chess can be baffling for the beginner, the rules of Draughts are more easily understood, but nevertheless both of them remain fascinating games for expert and novice alike. The topic under discussion now revolves around the board rather than the pieces on it, so, if you have such a board, you may care to get it out and study it. If not, you will just have to imagine an eight by eight board: don't worry about black and white squares, just stick to the eight by eight board, and we can continue.

After a quick calculation you might say that there are 64 squares on the board, and indeed there are 64 squares that are just one square in width and height. But what about the squares that are two squares in width and height, or three squares, or even eight squares? Of course, there is only one of the latter, but there are many more of the other sizes and all the intermediate ones. How many are there in total?

As if that were not puzzling enough, consider also that there are many, many more rectangles formed than there are squares. Rectangles can be two squares by one, or four squares by five, and there are far more of them than the more conventional squares.

So how many rectangles can be formed on a standard Chess board? and having discovered that one, what is the final total for the combined number of rectangles and squares?

Fortunately for us there are mathematical formulae for solving these particular riddles. However, if I were to simply tell you the answer for a square the size of a chess board, that would not tell you the solution for a nine by nine board, or an even greater fourteen by fourteen one. Rescue is at hand once more in the form of the accompanying program. This will allow you to solve the riddle of how many squares and rectangles there are for a whole range of squares from the simple two by two to the much more complex twenty by twenty.

Of course, you could use even larger numbers, although the square itself would not be drawn on the screen and the listing would need amending accordingly.

First, let us consider the formulae. If we have a variable 'A' equal to the size of the square (and thus 'A' would equal 8 for our Chess board problem), then the combined total of squares and rectangles that can be formed is given by the formula ((a*a+a) * (a*a+a))/4. This gives us the total number, and the individual number of squares is given by the formula (2*a*a*a+3*a*a+a)/6. From this we can of course work out the number of rectangles easily enough, but just for the sake of completeness the formula that itself gives us the number of rectangles not including squares is (3*a*a*a*a+ 2*a*a*a- 3*a*a- 2*a) / 12. A complicated one!

That brings us to the program listing, and a simple enough one it is too once we know the relevant formulae. Lines 10 and 20 serve to identify it, while lines 30 and 40 select our colour scheme and clear the screen for neatness. Lines 50 to 60 then allow the

user to input a number in the range 2 to 20, and could be amended if bigger squares were required, although you would then be advised not to attempt drawing them and skip straight to line 140.

Line 70 clears the screen again, lines 80 to 130 define and draw the square in its entirety, before we get to the actual calculations. After a brief statement in line 140, line 150 then works out the total number of squares and rectangles using the formula given earlier. Line 160 calculates the number of squares and the number of rectangles not including squares. That these formulae do actually give us the correct answer can soon be verified by drawing a trivial square two by two, for instance, which is easily calculated by the eye.

The answers are displayed by lines 170 to 190, and line 200 rounds everything off and prevents the program falling through to line 210 and producing a RETURN WITHOUT GOSUB error.

That is the program, those are the formulae, and an interesting little mathematical diversion they form as well. However, an article of this nature would not be complete without giving you some work to do as well. As we have seen, there are formulae for squares, but what about triangles? Imagine an equilateral triangle that is divided into eight smaller triangles along its base, eight along its two remaining sides, and the interior filled in with triangles of the same size. Rather like an eight by eight Chess board, we might be said to have an eight by eight by eight triangle. Remembering that triangles can be formed upside down as well as the right way up, what is the total number of triangles contained within the big triangle, noting that the big triangle as a whole adds one to the total, just as the entire Chess board adds one to the number of squares. Easy if you know the formula, but if you don't.....!!!

Switching The Bases

As human beings we are accustomed to counting in a numerical system that uses ten as its base, hence we arrive at what is called a decimal system. This is not particularly convenient for computers which are ideally suited to the so-called binary system.

As far as computers are concerned an electronic circuit can be either turned on or off, it only has two possible values associated with it, and, if we give those values the numerical associations of zero and one, we have the basis for this binary system.

Binary and decimal do not mix happily though, and so a third counting system comes into the reckoning. This is known as hexadecimal, using the number 16 as its base. Unfortunately the Arabic numerals from which our own derive only ascend as far as 9, using the symbols 0, 1, 2 and so on. Rather than inventing new symbols, hexadecimal uses letters of the alphabet. These letters are A, B, C, D, E and F, and this gives us hexadecimal numbers of the form 12FE for example.

In order to convert numbers from one numerical system to another, we need only know the base that is being used. For instance, the decimal number 1234 is easily translated as 4 plus 3 times 10 plus 2 times 10 to the power 2 plus 1 times 10 to the power of 3. The further left we go in the number, the greater the power of ten that is being used. To take our earlier hexadecimal example of 12FE, knowing that E is equivalent to the decimal number 14 and F is equivalent to the decimal number 15, we arrive at 14 plus 15 times 16 plus 2 times 16 to the power of 2 plus 1 times 16 to the power of 3, or 1790 in our more familiar decimal system.

Thus the basis for all counting systems rests in the base being used, and by multiplying that base number raised to the appropriate power by the number concerned, we can easily convert from one numerical base to another. To give another example, let us assume a numerical system that uses the base 7. An unusual choice, perhaps, but an interesting one nonetheless. If we look at a number of 1234 which was expressed using this base of 7, to convert it back into our own decimal system we have 4 plus 3 times 7 plus 2 times 7 to power 2 plus 1 times 7 to power 3. This equates to a decimal number of 466.

Now we arrive at the mathematical diversions. To begin with, we will be using a base of 8, a base of 5, and a base of 7. In other words, three different bases. What we want to find is a solution to the following equation, assuming the variable 'A' to be the number we are looking for: A (base 8) minus A (base 5) equals A (base 7). The same number, but using a different numerical base for each time that it is expressed. What is more, we want to find a four digit number that will obey the equation, and, to complicate matters still further, we would like to find a number which has its second and third digits equal. A number like 1221, for example, although that is not the answer.

Many such problems can be devised, and I shall leave you with one to ponder on at the end of this article. For now, let us consider the problem as it stands and the program listing that enables us to solve it. With the rules of the problem clearly expressed, and the method of translating numbers from one base to another already defined, the listing should pose no problems.

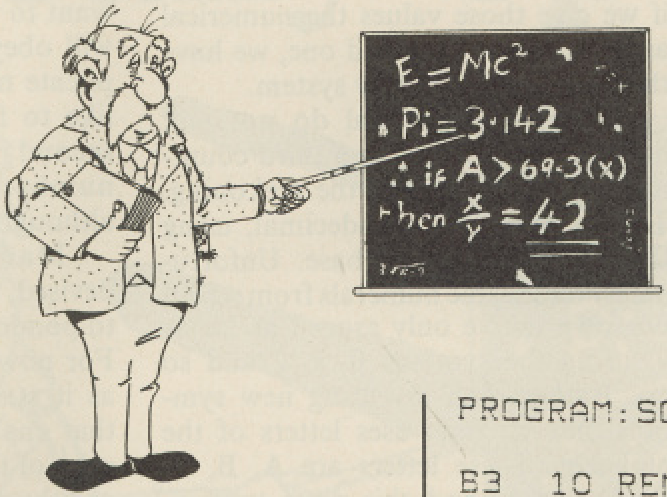
Line 10 serves to identify the program and its author before line 20 sets our colour scheme, while line 30 clears the screen. The loop in line 40 is set up to cover all possible four digit numbers, and with lines 50 and 60 we set up a string variable so that we can check for the presence of two equal digits in the second and third positions within those four digit numbers.

If those digits are not the same then line 70 sends us straight to line 100 to continue the search, ultimately concluding with that line if no such number is found. However, if the second and third digits are the same then we go in turn to lines 110, 130, 150 and 170 in order to convert that number to base 8, 5, 7 and 10 respectively, before going to line 190 and displaying the numbers on the screen to inform the user of the progress being made.

If the equation now holds true then line 90 tells us of the relevant number before halting the program with an END statement, otherwise we carry on with the loop in line 100.

The first number that conforms with our equation is the decimal number 1332. That is, 1332 in base 8 minus 1332 in base 5 equals 1332 in base 7. The decimal number 2664 also obeys the rules.

That is a simple example of the kind of problem that can be set when using a mixture of bases for counting purposes. Now consider a different sort of problem using the equation A (base x) equals A (base y) plus A (base z). If the numeric variable 'A' is representing the decimal number 1171, what are the values of x, y and z? In other words, what bases are we using??



PROGRAM:BASES

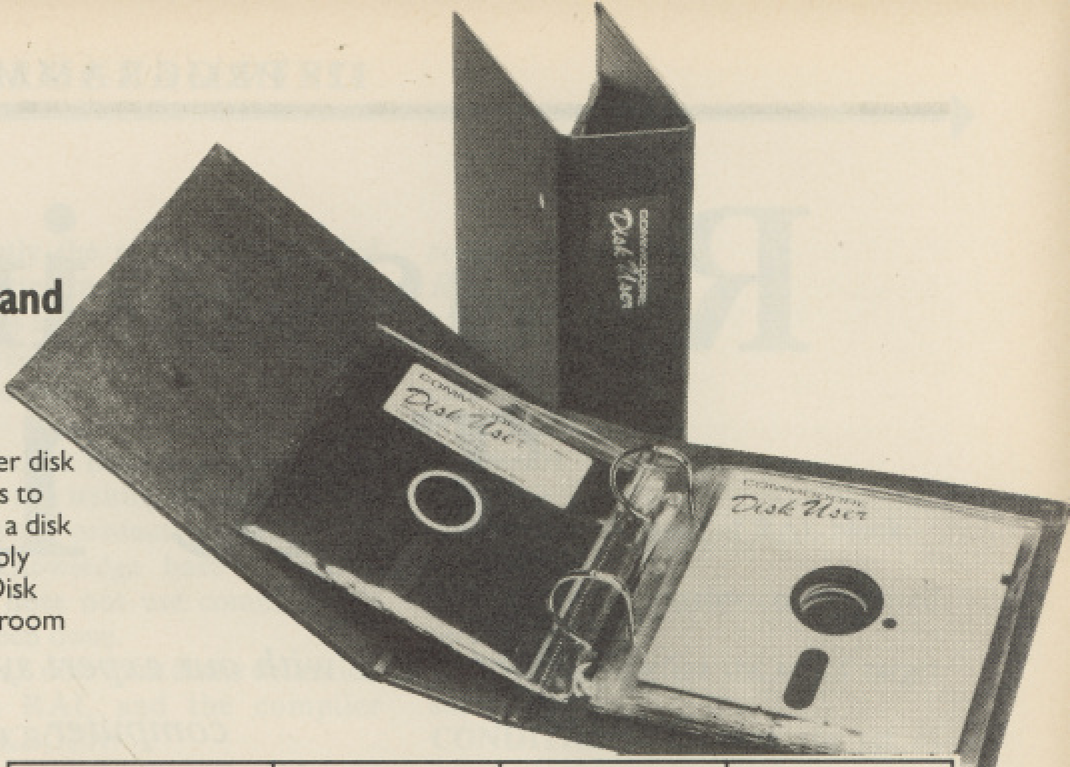
```
8D   10 REM BASES BY PETE GERRARD
B3   20 X=5328*10:POKE X,2:POKE X
     +1,0:PRINT CHR$(158)
03   30 PRINT CHR$(147)
1B   40 FOR I=1000 TO 9999
6C   50 A=I
3E   60 A$=MID$(STR$(A),2)
98   70 IF MID$(A$,2,1)<>MID$(A$,
     3,1) THEN 100
65   80 GOSUB 110:GOSUB 130:GOSUB
      150:GOSUB 170:GOSUB 190
25   90 IF A%-B%=C% THEN GOSUB 19
     0:PRINT:PRINT D$;" IS ONE PO
     SSIBLE ANSWER.":END
08   100 NEXT I:PRINT "NO NUMBER
     FOUND.":END
74   110 REM BASE 8
B7   120 A1=0:FOR J=1 TO 4:X$=MID
     $(A$,J,1):X=VAL(X$):A1=A1+X*
     8^(4-J):NEXT:A%=A1:RETURN
3D   130 REM BASE 5
0D   140 A1=0:FOR J=1 TO 4:X$=MID
     $(A$,J,1):X=VAL(X$):A1=A1+X*
     5^(4-J):NEXT:B%=A1:RETURN
4B   150 REM BASE 7
0E   160 A1=0:FOR J=1 TO 4:X$=MID
     $(A$,J,1):X=VAL(X$):A1=A1+X*
     7^(4-J):NEXT:C%=A1:RETURN
41   170 REM BASE 10
3B   180 A1=0:FOR J=1 TO 4:X$=MID
     $(A$,J,1):X=VAL(X$):A1=A1+X*
     10^(4-J):NEXT:D%=A1:RETURN
9D   190 REM DISPLAY NUMBERS
06   200 PRINT CHR$(147);
70   210 PRINT "BASE  8 =";A%
74   220 PRINT "BASE  5 =";B%
6B   230 PRINT "BASE  7 =";C%
2F   240 PRINT "BASE 10 =";D%:D$=
     MID$(STR$(D%),2)
74   250 RETURN
```

PROGRAM:SQUARES

```
B3   10 REM SQUARES BY PETE GERRA
     RD
4E   20 REM FOR THE COMMODORE 64
4C   30 X=5328*10:PRINT CHR$(147)
     :POKE X,2:POKE X+1,0:PRINT C
     HR$(158)
9D   40 PRINT CHR$(147)
3C   50 INPUT "SIZE OF SQUARE (E.
     G. 8 FOR 8 BY 8)[SPC6]BETWEE
     N 2 AND 20[SSPC]";A
A4   60 IF A<2 OR A>20 THEN 40
DB   70 PRINT CHR$(147)
88   80 A=A-1:A$="":B$="":C$=""
B6   90 FOR I=1 TO A:A$=A$+CHR$(1
     78):NEXT:A$=CHR$(176)+A$+CHR
     $(174)
F4   100 FOR I=1 TO A:B$=B$+CHR$(
     219):NEXT:B$=CHR$(171)+B$+CH
     R$(179)
58   110 FOR I=1 TO A:C$=C$+CHR$(
     177):NEXT:C$=CHR$(173)+C$+CH
     R$(189)
EA   120 REM STRINGS DEFINED
2B   130 PRINT A$:FOR I=1 TO A:PR
     INT B$:NEXT:PRINT C$
06   140 X=22:Y=0:GOSUB 210:PRINT
      "SQUARES & SQUARES"
F4   150 A=A+1:Y=Y+3:TNR=((A*A+A)
     *(A*A+A))/4
35   160 NS=(2*A*A*A+3*A*A+A)/6:N
     R=(3*A*A*A*A+2*A*A*A-3*A*A-2
     *A)/12
2D   170 GOSUB 210:PRINT "TOTAL S
     QUARES &":Y=Y+1:GOSUB 210:PR
     INT "RECTANGLES =";TNR
E6   180 Y=Y+2:GOSUB 210:PRINT "S
     QUARES[SPC4]=";NS
7D   190 Y=Y+2:GOSUB 210:PRINT "R
     ECTANGLES =";NR
A5   200 X=0:Y=20:GOSUB 210:END
75   210 PRINT CHR$(19);:FOR I=1
     TO Y:PRINT:NEXT:FOR I=1 TO X
     :PRINT CHR$(29);:NEXT:RETURN
```

Binders

Organise and protect your disk with Commodore Disk User disk binders and data disks.



Why not keep your Commodore Disk User program collection alongside your magazines in a stylish Disk User disk binder? The binder comes complete with 10 disk sleeves to organise and protect your program disks. Why not buy a disk binder to house all of your data disks? We can even supply Commodore Disk User data disks. The Commodore Disk User logo immediately identifies your disks and there's room to title them and document the disks details. Send for your disks and binders now!

Prices are as follows:

Commodore Disk User Binder £4.95, including 10 sleeves. Order code **BDYUI**

Commodore Disk User Binder with 10 sleeves and 10 disks, £9.95 Order code **BDYU2**

10 sleeves for insertion in binder, £1.50. Order code **BDS10**

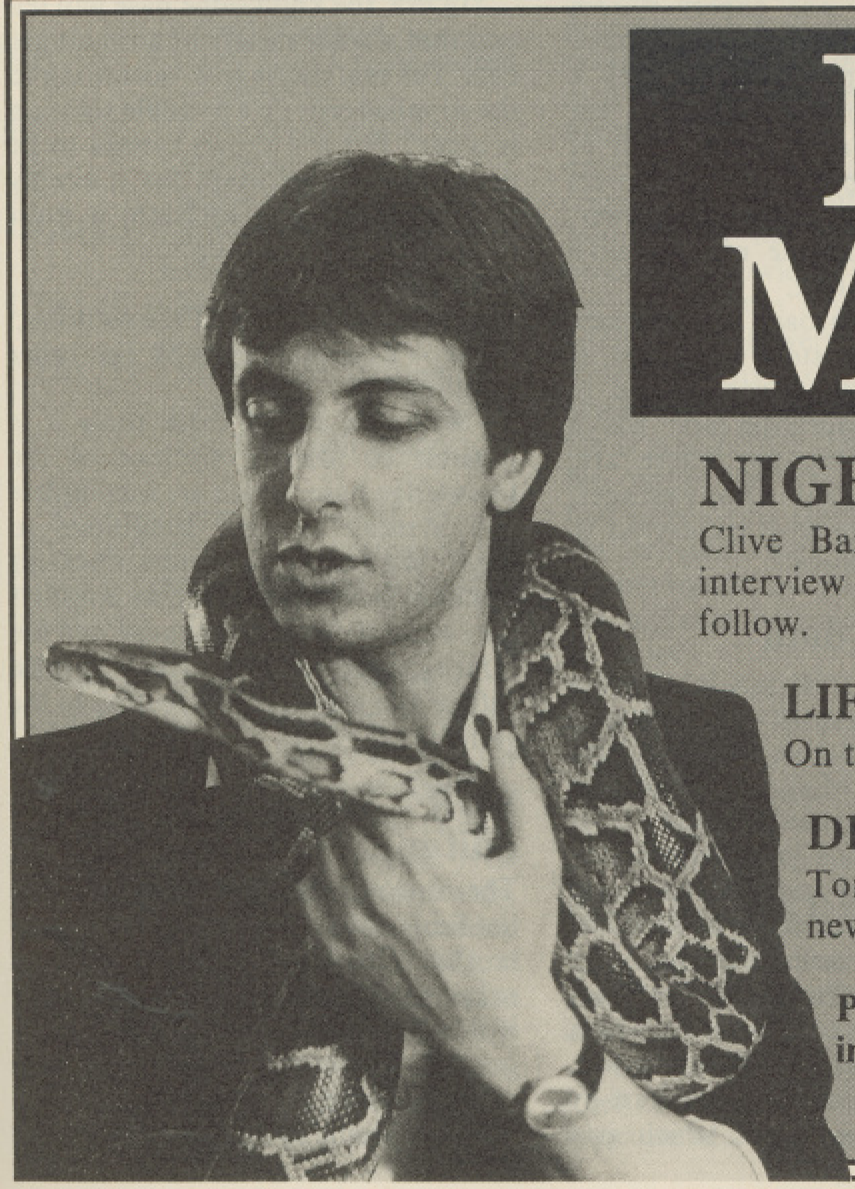
20 sleeves for inclusion in binder, £2.75. Order code **BDS20**

10 Commodore Disk User data disks, £5.95. Order code **BDD10**

PRODUCT NAME	ORDER CODE	QUANTITY	PRICE
Overseas postage add £1.00			
		TOTAL	

All orders should be sent to: YOUR COMMODORE, READERS SERVICES, ARGUS SPECIALIST PUBLICATIONS, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST. Please allow 28 days for delivery.

NEXT MONTH



NIGHTBREED

Clive Barker spills his guts in an exclusive interview about his new film and the game to follow.

LIFESTYLES 2000

On the spot report.

DESERT ISLAND DISKS

Tony Crowther is the first to contribute to our new regular spot.

Plus Role Playing Games, Play By Mail, Software in the Office...

Reasoning On The 128

We continue with our expert system for the 128 computer

By Paul Schofield

In the last installment of this series we introduced you to the idea of a simple expert system, from now on, we will look at the more sophisticated knowledge base which I introduced in that article. This system comprises of three programmes:

EDR – the source program editor
KBDEN – the knowledge base compiler
INQUIRE – the inference engine

Why three programs? The most important consideration is space available for data, which is increased by keeping the programs small. Speed of operation is also improved, as once the application has been debugged, it is only necessary to use the *INQUIRE* program which reads in the data in a compressed and preprocessed format. In this section, we will be concerned mainly with the editor, but first a quick overview of the whole system.

Defining a knowledge base

To create the knowledge base for the reason program, we first processed all the data that we had on the subject in question and arranged it into a tree structure. In the case of the very simple chemical analysis example, we ended up with a nice simple symetric tree, which therefore took an efficient path to any of the solutions. If we were to extend this, we would have to start using tests to identify particular chemicals and so the tree would no longer by symetric. This is similar to the car maintenance example, where it was suggested that the first questions might be:

fuel tank is empty?
battery is flat?

In this case, we pick these as starting questions, because they are easy to check and the produce a simple solution.

In terms of our expert system however, they represent a fundamentally different approach. We pick a solution to the problem, namely, put petrol in fuel tank and ask a set of questions to support it. If the answers are affirmative we have an answer, otherwise we take the next possible solution and test that.

This is the approach that our new knowledge base system will adopt. It is probably the most widely used in this type of application and has a number of advantages. There are, however, attendant disadvantages, and so the system must endeavour to take account of these.

Incomplete data

One of the first rules one learns as a programmer is, garbage in means garbage out, and this is particularly true of this type of expert system. To define your knowledge base, it is necessary to define your targets (the possible solution) and the conditions which must be satisfied for a particular target to be the solution. The order in which you define the targets also implicitly defines the order in which they are evaluated. This is good, because at the outset you can start by just entering conditions, which you know to be true about a target. The disadvantage is easily shown by an example. Suppose you write a knowledge base for animal identification. The first target is cat and the conditions are:

Has four legs?
Has a tail?

In use, the knowledge base will tell you that a wide variety of animals are cats. The real message is 'according to the supplied criteria a possible solution is a cat'. For the expert system to be useful it's not sufficient for it simply to stop at the first solution, it must provide to additional options:

Why – A justification of the solution,
More – Continue to look for other solutions.

With the aid of these tools it is possible to identify sections of the knowledge base, which are incomplete and identify suitable conditions to apply. We will consider such problems in a little more detail when we consider the source definition language and the inference engine, but this should suffice as a general introduction to the system.

EDR - The source file editor

The obvious question is why not use the Basic editor? One reason of course is that the Basic editor produces tokenized program files, but in general it is more convenient to use either a general purpose text editor or one specific to the particular application. *EDR* is specific to this application, but

would not require much modification to make a general purpose text editor. To run the editor, simply enter: run 'EDR' The editor initialises various variables and then displays the main options menu. Most of these are self explanatory, so we will just consider option 1, create/edit RAL source file. When this option is selected, a help bar is displayed at the top of the screen showing the functions assigned to the function keys. The remainder of the screen is either blank (create) or displays the start of the currently loaded file. The files produced are sequential files, so no line numbers are needed, just enter the text you require. The following special function keys are also supported:

CURSOR KEYS	: Move up/down/left/ right one character
DEL	: Delete character left of cursor
INST	: Insert blank character at cursor
F1	: Insert blank line
F2	: Delete current line
RETURN	: Move cursor to start of next line
F6	: Delete remainder of line to right of cursor
F4	: Copy line to paste buffer and delete it
F3	: Paste saved line at car
F5	: Move csr to top of file
F7	: Move csr to end of file
HOME	: Move csr to top left of screen
CTRL/U	: Move up 20 lines or to top of file
CTRL/D	: Move down 20 lines or to end of file
F8	: exit from edit mode

One automatic function of the editor is that when the bottom of the screen is reached, the text is scrolled up 10 lines.

Modyfying The Editor

The functions supported by the editor are all that you are likely to need for entering knowledge base definitions. For more general use a few modifications are likely to be needed. New edit functions can be added very easily. Add a test for the key in question between line 10300 and line 10500 and insert the corresponding subroutine somewhere between line 2000 and line 4999.

Two additional modifications may also be required:

1) Currently the number of lines of text is written at the start of the disk files. If this is considered undesirable, the disk read/write routines must be changed to detect eof.

2) Commodore Basic has trouble with commas when writing text to disk files unless special precautions are taken. As the knowledge base definition language does not use commas, this has not been done.

We shall now look at the definition language RAL and the compiler program KBGEN.

Having experimented with the REASON program, and typing in the EDR Editor. We will now look at the knowledge base defination. Later, we will look at the compiler program, KBGEN. First of all though, we will look at the defination language. As these 3 programs together represent the package Reason Version 3, I have called the language, Reason Application Language (RAL).

Before going into details though, a quick look at some of the objectives. Although KBGEN is technically a compiler, I prefer to think of it as a simple preprocessor. It does not generate machine code, but simply takes a description of data structures in a readable format and translates them into a set of arrays suitable for driving an inference engine. It was also considered important, that the compiler should be able to output the processed knowledge base on a different disk to the source file. This is adminstratively very convenient, but, if tedious disk swapping is to be avoided, requires that the entire source file is read before attempting to write the output file. This in turn dictates that large string storage requirements are provided and in order not to unduly restrict the size of applications the language becomes a little cryptic in places.

It is possible to change this approach, but I would recommend trying the existing system first, it's not really that difficult.

Structure of RAL

RAL is a block structured language like Pascal or C, but the syntax of the individual blocks is a bit more rigid.

Outside any of the blocks comment lines are permitted, so the basic structure of a RAL program is:

comments
RAL BLOCK
comments
:
comments
RAL BLOCK
comments

There are 5 types of RAL block:-

HELP – Initial instructions displayed to user of base.
TARGETS – Define the set of target solutions.
CONDITIONS – Define the set of questions the user is asked.
RELATIONS – Define set of rules for evaluating targets.
IMPLIES – Defines how additional conditions may be inferred from a users response.

The structure of each block is as follows:

BLOCK-HEADER
block statements
END [*comment*]

The square brackets [] *indicate optional elements.*

Comments

As mentioned previously, comments may be included anywhere outside a block, they have the format:

; [any text string]

It is important that the ';' is the first character on comment lines and these are simply ignored by the compiler. It would not be difficult to permit comments within blocks, but it would greatly slow down the compiler and is not very useful.

Help Section

The help section is optional, but should normally be included in all RAL definitions. It simply defines a number of lines of explanatory text that will be displayed to the user before any questions are asked. It may typically be used to display a title, take credit for creating the knowledge base and providing any necessary background information. It is not necessary to include any information on how to respond to reason questions as this is automatically output by the INQUIRE program. The format of the help section is:

HELP integer
help-text-lines
end [*HELP*]

where:
integer defines the number of lines of help information following,
help-text-lines are the help information in the format to be displayed.

Targets Section

The targets section defines the text strings describing the target solutions and is obviously mandatory. The format of the targets section is:

TARGETS integer
target-lines
END [*TARGETS*]

where:
'integer' defines the number of entries to be reserved for targets.
and *target-lines* have the format:

[index] target-text

where:
index is an optional target number reference (1,2,3 etc..)
target-text is the text that will be displayed to describe the solution.

Note

The purpose of integer is to define internal storage requirements and so must be the same as or greater than the number of following targets. The *index* is purely a comment, a bit like a Basic line number. It is used in later sections to identify specific targets and is therefore useful in large knowledge bases. The first index is always 1 and they are allocated consecutively.

Conditions Section

The conditions section defines the questions which will be asked to the user and is obviously mandatory. The format of the conditions section is:

CONDITIONS integer
condition-lines
END [*CONDITIONS*]

where:
integer defines the number of entries to be reserved for conditions.
condition-lines have the format:

[*index*] *condition-text*
where:
index is an optional condition number reference (1,2,3 etc..)
condition-text is the text of the questions displayed to the user.
integer and *index* are treated as for targets.

Relations Section

The relations section defines the rules for evaluating targets according to the answers to the condition questions. The results of the various conditions may be either
irrelevant
(must be true for target to be a solution) or *relevant* (must be false for target to be a solution).

By default, all conditions are considered irrelevant unless mentioned explicitly within a rule in the relations section. The format of the relations section is:

RELATIONS
relation-rules
END [*RELATIONS*]

where:
relation-rules have the format: *target-index operator condition-index-list*
where:
operator is '+' following conditions must be ture, or '-' following conditions must be false.
condition-index-list is a number of condition indices separated by spaces.

Implies Section

The implies section is optional. It defines the results of further conditions, which may be determined from the result of the current condition. The object of this section is to avoid asking questions whose results should already be known and to a lesser extent, avoid investigating impossible targets as quickly as possible. For example, if during the investigation of one target the question is asked "IS OBJECT BLUE?" and the answer is true, it is quite obvious that a later question "IS OBJECT RED?" must have the answer false. The implies section allows upto three aditional results to be implied for both a true or false answer to a particular condition. Its format is:

IMPLIES
implication-rules
END [*IMPLIES*]

where the format of implication rules is:

result-condition implied-condition [implied-condition [implied-condition]]

and the format of both result-condition and implied condition is:

'+/-' condition-index

As with relation rules + = true and - = false.

If that sounds a little confusing, then have a look at the example program in figure 1. Yes we are back with our mixed up chemical bottles again! With only 5 targets and 4 conditions this is very simple and it is easy to see how the relations and implies sections are used.

It is always best to work through the targets in the order they are defined as this is also the order in which they will be tested. In this case we check firstly for a strong acid. For this to be the case, blue litmus must be turned strongly red. This is a rather subjective question, so the simpler condition Blue Litmus Turned Red is considered first. In the relations section, we see to rules defined, the first saying that conditions 1 and 2 must be true and a second requiring conditions 3 and 4 to be false. It is quite obvious in fact that if 1 and 2 are true that 3 and 4 will be false, but this is defined in the implies section instead. The reason for doing it this way is that at some time the future it could be decided to reorganise to test targets in a different order. This becomes difficult if the rules of the relations section are incomplete.

KBGEN program

The structure of this program is quite simple. The main program reads in the RAL source file line by line and looks for the sections in the expected order. Within each section the lines are validated according to the defined syntax. Any errors encountered are reported and if the error is fatal the processing aborted. Standard subroutines are used for skipping spaces and finding and extracting various elements in the source lines. It should not be that difficult to make changes to the program if desired.

The most obvious change to consider is to open the output file on the same disk at the start of the program. As the format of the help, targets and condition sections is very simple, these can be written directly to the output file after being validated so the large string arrays are avoided. This saving

can then be used to allow targets and conditions to be referenced by an identifier rather than an index. This has the advantage of making it simpler to reorganise a knowledge base, but searching the look-up tables needed will make compilation much slower. It also makes the relations section longer and the lines of the implies section more difficult to parse. In principle there is no limit to the length of such identifiers, but in practice the longer the identifiers the shorter are the corresponding target and condition texts.

Thats it for this monghs excurskon into reasoning, the next and final part will deal with the Inference Engine INQUIRE and also a look at an application of this type of knowledge base system.

PROGRAM: EDR

```
10 GOTO 50000
40 REM ------------------------
-------------------------------
-----
50 REM EDITOR SUBROUTINES
60 REM ------------------------
-------------------------------
-----
99 REM PRINTABLE CHARACTER
100 IF LP>EF THEN EF=LP:IF LP>ML
 THEN EF=ML:ER=1:GOTO 1700
110 LC=LEN(T$(LP)):IF CP>LC THEN
 T$(LP)=T$(LP)+K$:ELSE T$(LP)=LE
FT$(T$(LP),CP-1)+K$+RIGHT$(T$(LP
),LC-CP)
120 PRINT K$;:CP=CP+1:IF CP=MC+1
 THEN CP=1:LP=LP+1:PRINT SN$;
130 IF LP>LL THEN LL=LL+1:TL=TL+
1:PRINT T$(LP);SL$;:RETURN
140 RETURN
150 REM ------------------------
-------------------------------
------
199 REM DELETE CHARACTER
200 LC=LEN(T$(LP)):IF CP=1 THEN
RETURN
210 T$(LP)=LEFT$(T$(LP),CP-2)+RI
GHT$(T$(LP),LC-CP+1)
220 PRINT DC$;:CP=CP-1:RETURN
250 REM ------------------------
-------------------------------
------
299 REM INSERT CHARACTER
300 LC=LEN(T$(LP)):IF CP>LC THEN
 RETURN
310 IF CP=1 THEN T$(LP)=SP$+T$(L
P):GOTO 330
320 T$(LP)=LEFT$(T$(LP),CP-1)+SP
$+RIGHT$(T$(LP),LC-CP+1)
330 LC=LC+1:IF LC>MC THEN T$(LP)
=LEFT$(T$(LP),MC)
340 PRINT SL$;EE$;T$(LP);SL$;:IF
 CP<>1 THEN FOR I=1 TO CP-1:PRIN
T CR$;:NEXT
350 RETURN
360 REM ------------------------
-------------------------------
------
399 REM INSERT LINE (F1)
400 IF LP>EF THEN RETURN
410 EF=EF+1:IF EF>ML THEN EF=ML:
ER=1:GOTO 1700
420 CP=1:FOR I=EF TO LP+1 STEP -
1:T$(I)=T$(I-1):NEXT:T$(LP)="":P
RINT SC$;
430 GOTO 1260
450 REM ------------------------
-------------------------------
------
499 REM DELETE LINE (F2)
500 EF=EF-1:FOR I=LP TO EF:T$(I)
=T$(I+1):NEXT:T$(EF+1)=""
510 PRINT SL$;DL$;:CP=1:RETURN
550 REM ------------------------
-------------------------------
------
599 REM RETURN
600 LC=LEN(T$(LP)):IF CP>LC THEN
 PRINT SP$;SN$;:ELSE PRINT MID$(
T$(LP),CP,1);SN$;
610 CP=1:LP=LP+1:IF LP<=LL THEN
RETURN
620 LL=LL+10:TL=TL+10:FOR I=LP T
O LL-1:PRINT T$(I);SN$;:NEXT
630 PRINT T$(LL);SL$;:FOR I=LP T
O LL-1:PRINT CU$;:NEXT:RETURN
650 REM ------------------------
-------------------------------
------
699 REM DELETE TO END OF LINE (F
6)
700 T$(LP)=LEFT$(T$(LP),CP-1):PR
INT EE$;:RETURN
750 REM ------------------------
-------------------------------
------
799 REM CUT LINE (F4)
800 T$(0)=T$(LP):GOTO 500
850 REM ------------------------
-------------------------------
------
899 REM PASTE LINE (F3)
900 IF LP>EF THEN EF=LP:T$(LP)=T
$(0):PRINT T$(LP);SL$;:RETURN
910 GOSUB 410:T$(LP)=T$(0):PRINT
 T$(LP);SL$;:RETURN
950 REM ------------------------
-------------------------------
------
999 REM CURSOR RIGHT
1000 IF CP=MC THEN RETURN
1010 IF CP>LEN(T$(LP)) THEN T$(L
P)=T$(LP)+SP$:PRINT SP$;:CP=CP+1
:RETURN
1020 PRINT MID$(T$(LP),CP,1);:CP
=CP+1:RETURN
1050 REM -----------------------
-------------------------------
-------
1099 REM CURSOR LEFT
1100 IF CP=1 THEN RETURN
1110 IF CP>LEN(T$(LP)) THEN PRIN
T SP$;CL$;CL$;:CP=CP-1:RETURN
1120 PRINT MID$(T$(LP),CP,1);CL$
;CL$;:CP=CP-1:RETURN
1150 REM -----------------------
-------------------------------
-------
1199 CURSOR UP
1200 IF LP=1 THEN RETURN
1210 IF LP=TL THEN 1250
1220 LC=LEN(T$(LP)):IF CP>LC THE
N PRINT SP$;CL$;:ELSE PRINT MID$
(T$(LP),CP,1);CL$;
1230 LC=LEN(T$(LP-1))+1:IF CP>LC
 THEN FOR I=CP TO LC+1 STEP-1:PR
INT CL$;:NEXT:CP=LC
1240 PRINT CU$;:LP=LP-1:RETURN
1250 CP=1:LP=LP-1:TL=TL-10:LL=LL
-10:PRINT SC$;:IF TL<1 THEN TL=1
:LL=24
1260 FOR I=TL TO LL-1:PRINT T$(I
);SN$;:NEXT
1270 PRINT T$(LL);HO$;:IF LP>TL
THEN FOR I=TL TO LP-1:PRINT CD$;
:NEXT
1280 RETURN
1290 REM -----------------------
-------------------------------
-------
1299 REM CURSOR DOWN
1300 IF LP=ML THEN RETURN
1310 IF LP=LL THEN GOTO 600
1320 LC=LEN(T$(LP)):IF CP>LC THE
N PRINT SP$;CL$;:ELSE PRINT MID$
(T$(LP),CP,1);CL$;
1330 LC=LEN(T$(LP+1))+1:IF CP>LC
 THEN FOR I=CP TO LC+1 STEP-1:PR
INT CL$;:NEXT:CP=LC
1340 PRINT CD$;:LP=LP+1:RETURN
1350 REM -----------------------
-------------------------------
-------
1399 REM TOP OF FILE (F5)
1400 CP=1:LP=1:TL=1:LL=24:PRINT
SC$;:GOTO 1260
1450 REM -----------------------
-------------------------------
-------
1499 REM BOTTOM OF FILE (F7)
1500 CP=1:LP=EF:LL=EF:TL=EF-23:P
RINT SC$;:IF TL<1 THEN TL=1:LL=2
4
1510 IF EF>TL THEN FOR I=TL TO E
F-1:PRINT T$(I);SN$;:NEXT
1520 PRINT T$(EF);SL$;:RETURN
1550 REM -----------------------
-------------------------------
-------
1599 REM HOME
1600 LC=LEN(T$(LP)):IF CP>LC THE
N PRINT SP$;CL$;:ELSE PRINT MID$
(T$(LP),CP,1);CL$;
1610 CP=1:LP=TL:PRINT HO$;:RETUR
N
1650 REM -----------------------
-------------------------------
-------
1699 REM EXIT FROM EDIT MODE (F8
)
1700 PRINT SC$;:EX=1:FL=1:RETURN
1750 REM -----------------------
-------------------------------
-------
1799 REM PAGE UP (^U)
1800 CP=1:LP=LP-20:TL=TL-20:LL=L
L-20:PRINT SC$;:IF TL<1 THEN TL=
1:LL=24
1810 IF LP<1 THEN LP=1
1820 GOTO 1260
```

```
1850 REM ----------------------
------------------------------
-------
1899 REM PAGE DOWN (^D)
1900 CP=1:LP=LP+20:TL=TL+20:LL=L
L+20:PRINT SC$;:IF LP>EF THEN LP
=EF
1910 IF TL>EF THEN TL=EF-10:LP=E
F:LL=TL+23
1920 IF TL<1 THEN TL=1:LL=24
1930 IF LL>ML THEN LL=ML:TL=LL-2
3
1940 GOTO 1260
1950 REM ----------------------
------------------------------
-------
1999 REM INSERT NEW FUNTION SUBR
OUTINES HERE
4999 REM GENERAL DISK ACCESS SUB
ROUTINE
5000 PRINT "CURRENT DISK IS : ■"
;CD$:SLOW:CATALOG:FAST
5010 PRINT "■■INSERT DATA DISK A
ND PRESS ANY KEY TO CONTINUE.":P
RINT
5020 GETKEY K$
5030 F$="":PRINT "DATA DISK DIRE
CTORY :■":PRINT:SLOW:CATALOG:FAS
T:PRINT"■":INPUT "NAME OF RAL SO
URCE FILE [12 CHARS MAX] ";F$
5040 IF LEN(F$)>12 THEN F$=LEFT$
(F$,12)
5050 F$=F$+".RAL":RETURN
5060 REM ----------------------
------------------------------
---
9999 REM START OF CREATE EDIT
10000 WINDOW0,0,79,24:PRINTSC$;"
■ ■F1■ INS LN ■F2■ DEL LN ■F3■
PASTE LN ■F4■ CUT LN ■F5■ TOF ■F
6■ DEL EOL ■F7■ EOF ■F8■ EXIT ■
":WINDOW2,1,77,24
10010 COLOR 5,1:PRINT SC$;:TL=1:
LP=1:CP=1:LL=24:EX=0:IF FL=0 THE
N EF=0:GOTO 10100
10020 FOR I=TL TO LL-1:PRINT T$(
I);SN$;:NEXT:PRINT T$(LL);HO$;
10099 REM EDITOR MAIN LOOP
10100 DO
10110 IF CP>LEN(T$(LP))THEN PRIN
T RN$;SP$;RF$;CL$;:GETKEY K$:I=A
SC(K$):ELSE:PRINT RN$;MID$(T$(LP
),CP,1);CL$;RF$;:GETKEY K$:I=ASC
(K$)
10120 IF I>31 AND I<94 THEN GOSU
B 100:GOTO 10500
10130 IF I=20 THEN GOSUB 200:GOT
O 10500
10140 IF I=148 THEN GOSUB 300:GO
TO 10500
10150 IF I=133 THEN GOSUB 400:GO
TO 10500
10160 IF I=137 THEN GOSUB 500:GO
TO 10500
10170 IF I=13 THEN GOSUB 600:GOT
O 10500
10180 IF I=139 THEN GOSUB 700:GO
TO 10500
10190 IF I=138 THEN GOSUB 800:GO
TO 10500
10200 IF I=134 THEN GOSUB 900:GO
TO 10500
10210 IF I=29 THEN GOSUB 1000:GO
TO 10500
10220 IF I=157 THEN GOSUB 1100:G
OTO 10500
10230 IF I=145 THEN GOSUB 1200:G
OTO 10500
10240 IF I=17 THEN GOSUB 1300:GO
TO 10500
10250 IF I=135 THEN GOSUB 1400:G
OTO 10500
10260 IF I=136 THEN GOSUB 1500:G
OTO 10500
10270 IF I=19 THEN GOSUB 1600:GO
TO 10500
10280 IF I=140 THEN GOSUB 1700:G
OTO 10500
10290 IF I=21 THEN GOSUB 1800:GO
TO 10500
10300 IF I=4 THEN GOSUB 1900:GOT
O 10500
10310 REM NEW FUNCTIONS CAN BE A
DDED HERE
10490 REM IGNORE ANY OTHER CHARA
CTERS
10500 LOOP UNTIL EX<>0
10510 GOTO 50100
19999 REM LOAD RAL SOURCE FILE
20000 PRINT SC$;RN$;CD$;" LOAD R
AL SOURCE FILE ";RF$;CD$
20010 IF FL<>0 THEN PRINT "■FILE
 ALREADY LOADED - PRESS ANY KEY
FOR MAIN MENU.":GETKEY K$:GOTO 5
0100
20020 GOSUB 5000
20030 SLOW:DOPEN#1,""+F$,D0,U8,R
:IF DS<>0 THEN PRINT "■■";DS$;"
- ANY KEY FOR MAIN MENU.":GETKEY
 K$:DCLOSE#1:GOTO 50100
20040 INPUT#1,EF
20050 FOR I=1 TO EF:INPUT#1,T$(I
):NEXT
20060 DCLOSE#1:FL=1:GOTO 50100
29999 REM SAVE RAL SOURCE FILE
30000 PRINT SC$:PRINT RN$;" SAVE
 RAL SOURCE FILE ";RF$;CD$:IF FL
=0 THEN PRINT "■ NO FILE IN TEXT
 BUFFER";CD$:GOTO 30070
30010 INPUT "ARE YOU SURE [Y/N]
";K$:PRINT
30020 IF K$<>"Y" THEN 50100
30030 GOSUB 5000
30040 SLOW:DOPEN#1,""+F$,D0,U8,W
:IF DS=0 THEN EX%=0:GOTO 35000
30050 IF DS=63 THEN DCLOSE#1:GOT
O 30090
30060 PRINT "■■ DISK ERROR ■■ ";
DS$:DCLOSE#1
30070 FAST:PRINT "PRESS ANY KEY
FOR MAIN MENU."
30080 GETKEY K$:GOTO 50100
30090 EX%=1:INPUT "■ SOURCE FILE
 EXISTS ■ - REPLACE [Y/N] ";K$
30100 IF K$<>"Y" THEN FAST:GOTO
50100
30110 DOPEN#1,"@"+F$,D0,U8,W:IF
DS<>0 THEN 30060
35000 PRINT#1,EF
35010 FOR I=1 TO EF:PRINT#1,T$(I
):NEXT
35020 DCLOSE#1:FAST
35030 PRINT "SOURCE FILE SAVED -
 PRESS ANY KEY FOR MAIN MENU.":G
ETKEY K$:GOTO 50100
39999 REM PRINT FILE
40000 PRINT SC$;CD$;RN$;" PRINT
FILE ";RF$;CD$
40010 IF FL=0 THEN PRINT "■NO FI
LE LOADED IN TEXT BUFFER - PRESS
 ANY KEY FOR MAIN MENU.":GETKEY
K$:GOTO 50100
40020 OPEN 4,4:IF ST<>0 THEN CLO
SE 4:PRINT "DEVICE ERROR - PRESS
 ANY KEY FOR MAIN MENU":GETKEY K
$:GOTO 50100
40030 FOR I=1 TO EF:PRINT#4,T$(I
):NEXT:PRINT#4,CHR$(13):PRINT#4,
"NUMBER OF LINES =";EF
40040 CLOSE 4:GOTO 50100
44999 REM CLEAR EDITOR TEXT BUFF
ER
45000 COLOR 5,3:CHAR 1,10,22,"AR
E YOU SURE [Y/N] : "
45010 GETKEY K$:PRINT K$;CL$;:IF
 K$<>"Y" AND K$<>"N" THEN 45010
45020 IF K$="N" OR FL=0 THEN 501
00
45030 FL=0:FOR I=0 TO EF+1:T$(I)
="":NEXT:EF=0:GOTO 50100
47999 REM FORMAT DATA DISK
48000 COLOR 5,3:PRINT SC$;:PRINT
 CD$;RN$;" FORMAT DATA DISK ";RF
$;SN$;CD$;
48010 INPUT "ARE YOU SURE [Y/N]
";K$:PRINT
48020 IF K$<>"Y" THEN 50100
48030 PRINT "INSERT BLANK DATA D
ISK IN DRIVE."
48040 INPUT "ENTER DISK NAME [12
 CHARS MAX] ";K$
48050 IF LEN(K$)>12 THEN K$=LEFT
$(K$,12)
48060 SLOW:K$="N0:"+K$+".RAL,KB"
:OPEN 1,8,15,K$:CLOSE 1:IF DS=0
THEN CATALOG:ELSE PRINT RN$;" ";
DS$;" ";RF$
48070 FAST:PRINT:INPUT "ANOTHER
DISK [Y/N] ";K$
48080 IF K$="Y" THEN PRINT:GOTO
48040
48090 GOTO 50100
49999 REM INITIALISATION SECTION
50000 ML=1500:MC=75:TL=1:LL=24:L
P=1:CP=1:EF=0:FL=0:LC=0:EX=0:EN=
0
50010 CL$="■":CR$=CHR$(29):CD$=C
HR$(17):CU$=CHR$(145):HO$=CHR$(1
9)
50020 DC$=CHR$(20):IN$=CHR$(148)
:SC$=CHR$(147):SP$=" "
50030 EE$=CHR$(27)+"Q":SL$=CHR$(
27)+"J":ES$=CHR$(27)+"@":EL$=CHR
$(27)+"K"
50040 SN$=SL$+CD$:RN$="■":RF$="■
":DL$=CHR$(27)+"D":DIM T$(1800)
50050 KEY 1,CHR$(133):KEY 2,CHR$
(137):KEY 3,CHR$(134):KEY 4,CHR$
(138):KEY 5,CHR$(135):KEY 6,CHR$
(139):KEY 7,CHR$(136):KEY 8,CHR$
(140)
50099 REM MAIN MENU
50100 WINDOW 0,0,79,24:COLOR 6,1
6:COLOR 5,7:PRINT SC$;:CHAR 1,10
,0,"        REASON APLICATION
LANGUAGE EDITOR  (EDR)          "
,1:PRINT CHR$(142);CHR$(11);
50110 WINDOW 2,1,77,24:CHAR 1,26
,2,"■ ■ OPTION  MENU  ■ ■",1
50120 CHAR 1,20,5,"■■ 1 ■■ CREAT
E / EDIT RAL SOURCE FILE"
50130 CHAR 1,20,7,"■■ 2 ■■ LOAD
RAL SOURCE FILE"
50140 CHAR 1,20,9,"■■ 3 ■■ SAVE
RAL SOURCE FILE"
50150 CHAR 1,20,11,"■■ 4 ■■ PRIN
T RAL SOURCE FILE"
50160 CHAR 1,20,13,"■■ 5 ■■ CLEA
R EDITOR TEXT BUFFER"
50170 CHAR 1,20,15,"■■ 6 ■■ FORM
```

```
AT DATA DISK"
50180 CHAR 1,20,17,"■ 7 ■ EXIT
"
50190 CHAR 1,20,19,"■ SELECT OPT
ION BY NUMBER  >> ■  ■"
50200 PRINT CHR$(143);:GETKEY K$
:PRINT K$;:I=ASC(K$)-48:SLEEP 1:
IF I<1 OR I>7 THEN PRINT CL$;:GO
TO 50180
50210 ON I GOTO 10000,20000,3000
0,40000,45000,48000
50220 WINDOW 0,0,79,24,1
50230 PRINT "■ WARNING ■ - SOME
SYSTEM DEFAULTS HAVE BEEN CHANGE
D !!"
50240 END
```

PROGRAM: KBGEN

```
10 FAST
20 SC$=CHR$(147):RN$="■":RF$="■"
:CD$=CHR$(17):II=0
30 GOTO 1000
40 REM DISCARD IDENTIFIER NUMBER
 AND SPACES
50 DO WHILE (LEFT$(L$,1)=" ") OR
 (VAL(LEFT$(L$,1))<>0) OR LEFT$(
L$,1)="0"
60 L$=RIGHT$(L$,LEN(L$)-1)
70 LOOP
80 RETURN
90 REM SKIP ANY COMMENT LINES
100 DO
110 INPUT#1,L$:LC=LC+1
120 LOOP UNTIL (LEFT$(L$,1)<>";"
) OR (LC>EF)
130 IF LC<=EF GOTO 160
140 PRINT "■ SYNTAX ERROR ■ UNE
XPECTED EOF AT LINE";LC
150 PRINT "■ PROGRAM ABORTED. ■
":END
160 RETURN
170 REM SKIP LEADING BLANKS
180 DO WHILE (LEFT$(L$,1)=" ")
190 L$=RIGHT$(L$,LEN(L$)-1)
200 LOOP
210 RETURN
220 REM EXTRACT NUMBER
230 N$=""
240 DO WHILE VAL(LEFT$(L$,1))<>0
 OR LEFT$(L$,1)="0"
250 N$=N$+LEFT$(L$,1):L$=RIGHT$(
L$,LEN(L$)-1)
260 LOOP
270 RETURN
280 REM FIND + OR -
290 DO
300 S$=LEFT$(L$,1):L$=MID$(L$,2)
310 LOOP UNTIL S$="+" OR S$="-"
OR LEN(L$) = 0
320 S=0:IF S$="-" THEN S=1
330 RETURN
340 REM FIND >
350 DO
360 S$=LEFT$(L$,1):L$=RIGHT$(L$,
LEN(L$)-1)
370 LOOP UNTIL S$=">" OR LEN(L$)
=1
380 IF LEN(L$) > 1 THEN RETURN
390 PRINT "■ SYNTAX ERROR ■" '>
' EXPECTED IN LINE";LC:PRINT E$
400 PRINT "■ PROGRAM ABORTED. ■
":END
499 REM WARNING - SHORT SECTION
500 PRINT"■ WARNING ■ SECTION S
HORTER THAN DIMENSIONED SIZE.■"
510 RETURN
1000 WINDOW 0,0,79,24:COLOR 6,16
:COLOR 5,7:PRINT SC$;:CHAR 1,10,
0,"              REASON KNOWLEDGE BA
SE GENERATOR (KBGEN)            ",
1:PRINT CHR$(142);CHR$(11);
1010 WINDOW 2,1,77,24:HN=0
1020 PRINT SC$;RN$;CD$;"■ SELECT
 RAL SOURCE FILE ■";RF$;CD$
1030 PRINT "CURRENT DISK IS : ■"
;CD$:SLOW:CATALOG:FAST
1040 PRINT "■INSERT RAL SOURCES
 DISK AND PRESS ANY KEY TO CONTI
NUE.":PRINT
1050 GETKEY Y$
1060 F$="":K$="":PRINT "DATA DIS
K DIRECTORY : ■":PRINT:SLOW:CATA
LOG:FAST:PRINT"■":INPUT"NAME OF
RAL SOURCE FILE [12 CHARS MAX] "
;F$
1070 IF LEN(F$)>12 THEN F$=LEFT$
(F$,12)
1080 K$=F$:F$=F$+".RAL":K$=K$+".
RKB"
1090 SLOW:DOPEN#1,""+F$,D0,U8,R:
IF DS<>0 THEN PRINT"■";DS$;" ■
PROGRAM ABORTED. ■":DCLOSE#1:EN
D
1100 INPUT#1,EF:FAST:LC=0
1110 PRINT SC$;RN$;CD$;"■ READIN
G RAL SOURCE FILE ■";RF$;CD$
1120 REM SKIP ANY COMMENT LINES
1130 GOSUB 100
1140 REM HELP OR TARGETS SECTION
 FOUND
1150 IF LEFT$(L$,7) = "TARGETS"
GOTO 1390
1160 IF LEFT$(L$,4) = "HELP" GOT
O 1210
1170 PRINT "■ SYNTAX ERROR ■ HE
LP OR TARGETS EXPECTED IN LINE";
LC:PRINT L$
1180 PRINT "■ PROGRAM ABORTED.
-"
1190 END
1200 REM READ HELP TEXTS
1210 HN=VAL(RIGHT$(L$,LEN(L$)-4)
):PRINT "READING HELP SECTION AT
 LINE";LC
1220 SS=0:HC=0:DIM H$(HN)
1230 DO
1240 INPUT#1,L$:LC=LC+1
1250 IF LEFT$(L$,3)="END" THEN H
N=HC:SS=1:GOSUB 500:GOTO 1270
1260 GOSUB 50:HC=HC+1:H$(HC) = L
$
1270 LOOP UNTIL (HC=HN) OR (LC=E
F)
1280 IF LC<=EF GOTO 1310
1290 PRINT "■ SYNTAX ERROR ■ UN
EXPECTED EOF AT LINE";LC
1300 PRINT "■ PROGRAM ABORTED.
-":END
1310 IF SS=1 GOTO 1350
1320 INPUT#1,L$:LC=LC+1:IF LEFT$
(L$,3)="END" GOTO 1350
1330 PRINT "■ SYNTAX ERROR ■ EN
D EXPECTED IN LINE";LC:PRINT L$
1340 PRINT "■ PROGRAM ABORTED.
-":END
1350 GOSUB 100:IF LEFT$(L$,7)="T
ARGETS" GOTO 1390
1360 PRINT "■ SYNTAX ERROR ■ TA
RGETS EXPECTED IN LINE";LC:PRINT
 L$
1370 PRINT "■ PROGRAM ABORTED.
-":END
1380 REM READ TARGET TEXTS
1390 TN=VAL(RIGHT$(L$,LEN(L$)-7)
):PRINT "READING TARGETS SECTION
 AT LINE";LC
1400 SS=0:TC=0:DIM T$(TN)
1410 DO
1420 INPUT#1,L$:LC=LC+1
1430 IF LEFT$(L$,3)="END" THEN T
N=TC:SS=1:GOSUB 500:GOTO 1450
1440 GOSUB 50:TC=TC+1:T$(TC) = L
$
1450 LOOP UNTIL (TC=TN) OR (LC=E
F)
1460 IF LC<=EF GOTO 1490
1470 PRINT "■ SYNTAX ERROR ■ UN
EXPECTED EOF AT LINE";LC
1480 PRINT "■ PROGRAM ABORTED.
-":END
1490 IF SS=1 THEN GOTO 1530
1500 INPUT#1,L$:LC=LC+1:IF LEFT$
(L$,3)="END" GOTO 1530
1510 PRINT "■ SYNTAX ERROR ■ EN
D EXPECTED IN LINE";LC:PRINT L$
1520 PRINT "■ PROGRAM ABORTED.
-":END
1530 GOSUB 100:IF LEFT$(L$,10)="
CONDITIONS" GOTO 1570
1540 PRINT "■ SYNTAX ERROR ■ CO
NDITIONS EXPECTED IN LINE";LC:PR
INT L$
1550 PRINT "■ PROGRAM ABORTED.
-":END
1560 REM READ CONDITION TEXTS
1570 CN=VAL(RIGHT$(L$,LEN(L$)-10
)):PRINT "READING CONDITIONS SEC
TION AT LINE";LC
1580 SS=0:CC=0:DIM C$(CN)
1590 DO
1600 INPUT#1,L$:LC=LC+1
1610 IF LEFT$(L$,3)="END" THEN C
N=CC:SS=1:GOSUB 500:GOTO 1630
1620 GOSUB 50:CC=CC+1:C$(CC) = L
$
1630 LOOP UNTIL (CC=CN) OR (LC=E
F)
1640 IF LC<=EF GOTO 1670
1650 PRINT "■ SYNTAX ERROR ■ UN
EXPECTED EOF AT LINE";LC
1660 PRINT "■ PROGRAM ABORTED.
-":END
1670 IF SS=1 GOTO 1710
1680 INPUT#1,L$:LC=LC+1:IF LEFT$
(L$,3)="END" GOTO 1710
1690 PRINT "■ SYNTAX ERROR ■ EN
D EXPECTED IN LINE";LC:PRINT L$
1700 PRINT "■ PROGRAM ABORTED.
-":END
1710 GOSUB 100:IF LEFT$(L$,9)="R
ELATIONS" GOTO 1750
1720 PRINT "■ SYNTAX ERROR ■ RE
LATIONS EXPECTED IN LINE";LC:PRI
NT L$
1730 PRINT "■ PROGRAM ABORTED.
-":END
1740 REM READ RELATIONSHIPS
1750 DIM R%(TN,CN):FOR I=1 TO TN
:FOR J=1 TO CN:R%(I,J)=0:NEXTJ:N
EXTI:D=0:PRINT "READING RELATION
S SECTION AT LINE";LC
1760 DO
1770 INPUT#1,L$:LC=LC+1:T=0
1780 IF LEFT$(L$,3)="END" THEN D
=1:GOTO 1890
1790 GOSUB 180:GOSUB 230
1800 TV=VAL(N$)
```

```
1810 GOSUB 180
1820 A=-1
1830 IF LEFT$(L$,1)="+" THEN A=1
1840 L$=RIGHT$(L$,LEN(L$)-1):GOSUB 180
1850 DO
1860 GOSUB 230:CV=VAL(N$):R%(TV,CV)=A
1870 GOSUB 180
1880 LOOP UNTIL LEN(L$)<=0
1890 LOOP UNTIL (D=1) OR (LC=EF)
1900 IF D=1 GOTO 1930
1910 PRINT " WARNING  UNEXPECTED END OF RELATIONS, INCOMPLETE ? ":GOTO 2170
1920 REM IMPLIES SECTION IS OPTIONAL - NO ERROR IF MISSING
1930 DO
1940 INPUT#1,L$:LC=LC+1
1950 LOOP UNTIL (LEFT$(L$,1)<>";") OR (LC>EF)
1960 IF LC>EF THEN 2170
1970 IF LEFT$(L$,7) = "IMPLIES" GOTO 2010
1980 PRINT " SYNTAX ERROR  IMPLIES EXPECTED IN LINE";LC:PRINT L$
1990 PRINT " REMAINING LINES IGNORED. ":GOTO 2170
2000 REM PROCESS IMPLICATION LINES - MAX 3 / TRUE OR FALSE ANSWER
2010 II=1:DIM I%(CN,1,3):FOR I=1 TO CN:FOR J=0 TO 1:I%(I,J,0)=0:NEXT:NEXT:D=0:PRINT "READING IMPLIES SECTION AT LINE";LC
2020 DO
2030 INPUT#1,L$:LC=LC+1
2040 IF LEFT$(L$,3)="END" THEN D=1:GOTO 2130
2050 GOSUB 290:A=S:GOSUB 230:IC=VAL(N$)
2060 GOSUB 350
2070 DO UNTIL LEN(L$)<=0
2080 GOSUB 180:GOSUB 290:IF LEN(L$)=0 GOTO 2120
2090 GOSUB 230:I%(IC,A,0)=I%(IC,A,0)+1:IP=VAL(N$):IF S=1 THEN IP=-IP
2100 IF I%(IC,A,0)<=3 THEN I%(IC,A,I%(IC,A,0))=IP:GOTO 2120
2110 PRINT " WARNING  MORE THAN 3 IMPLIES FOR CONDITION";IC;" EXTRA IGNORED. "
2120 LOOP
2130 LOOP UNTIL D=1 OR LC=EF
2140 IF D=1 GOTO 2170
2150 PRINT " WARNING  UNEXPECTED END OF IMPLICATIONS - INCOMPLETE ? "
2160 REM CLOSE SOURCE FILE
2170 DCLOSE#1:PRINT CD$;" END OF SOURCE FILE  PRESS ANY KEY TO CONTINUE.":GETKEY Y$
2180 REM INPUT FILE READ, WRITE KNOWLEDGE BASE FILE
2190 PRINT SC$;RN$;CD$;" WRITING KNOWLEDGE BASE FILE ";RF$;CD$
2200 PRINT "CURRENT DISK IS : ";CD$:SLOW:CATALOG:FAST
2210 PRINT " INSERT KNOWLEDGE BASE DISK AND PRESS ANY KEY TO CONTINUE.":PRINT
2220 GETKEY Y$
2230 SLOW:DOPEN#1,""+K$,D0,U8,W:IF DS=0 GOTO 2300
2240 IF DS=63 GOTO 2270
2250 PRINT " DISK ERROR  ";DS$
2260 DCLOSE#1:PRINT "PROGRAM ABORTED.":END
2270 DCLOSE#1:INPUT " KNOWLEDGE BASE FILE EXISTS  - REPLACE [Y/N] ";Y$
2280 IF Y$<>"Y" AND Y$<>"!" GOTO 2260
2290 DOPEN#1,"@"+K$,D0,U8,W:IF DS<>0 GOTO 2250
2300 REM WRITE SIZES AND THEN DATA ARRAYS
2310 PRINT#1,HN:PRINT#1,CN:PRINT#1,TN:PRINT#1,II
2320 IF HN>0 THEN FOR I=1 TO HN:PRINT#1,H$(I):NEXT
2330 FOR I=1 TO CN:PRINT#1,C$(I):NEXT
2340 FOR I=1 TO TN:PRINT#1,T$(I):NEXT
2350 FOR I=0 TO TN:FOR J=0 TO CN:PRINT#1,R%(I,J):NEXT:NEXT
2360 IF II=1 THEN FOR I=0 TO CN:FOR J=0 TO 1:FOR K=0 TO 3:PRINT#1,I%(I,J,K):NEXT:NEXT:NEXT
2370 DCLOSE#1
2380 PRINT CD$;" KNOWLEDGE BASE COMPILED.  PRESS ANY KEY TO EXIT."
2390 GETKEY Y$:WINDOW 0,0,79,24:PRINT SC$
```

PrintFx Correction

The Gremlins that caused havoc with the listing have been thwarted at last.

By Paul Eves

In the May edition of Your Commodore we published a Print Special Effects program for the Plus4.

Unfortunately, those dreaded gremlins reared their ugly heads and made a mess of the listings. You will no doubt have noticed some rather strange keywords and symbols. This was caused by the ROM in the printer having a day off.

Here then, for your pleasure, are the corrected listings. You will notice that listing 1 does not have the data statements on the end. (There is no point in re-typing these as they were correct originally).

We apologise for this error and hope that it will not spoil your enjoyment of this excellent program.

```
10 REM ***************************
**
20 REM *
 *
30 REM * PRINTFX +4 BASIC LOADER
 *
40 REM *
 *
50 REM ***************************
**
60 REM
70 GRAPHIC 1:GRAPHIC 0
80 PRINT CHR$(27)"RPRINTFX +4 BA
SIC LOADER PROGRAM"
90 PRINT"▣WRITTEN DECEMBER '88 B
Y M.R EVERINGHAM"
100 PRINT"▣
                    "
110 AD%=4097:FOR LI=1000 TO 3550
 STEP 10
120 PRINT"▣▣▣▣▣STORING DATA LIN
E"LI
130 CH%=0:FOR BY=0 TO 7
140 READ DA%:CH%=CH%+DA%
150 IF DA%<0 OR DA%>255 THEN PRI
NT"▣INVALID NUMBER IN LINE"LI:EN
D
160 POKE AD%+BY,DA%:NEXT BY
170 READ VR%:IF CH%<>VR% THEN PR
INT"▣CHECKSUM ERROR IN LINE"LI:E
ND
180 AD%=AD%+8:NEXT LI
190 PRINT"▣STORAGE COMPLETE - DA
TA 100% CORRECT"
200 PRINT"▣DO YOU WANT TO USE [T
]APE OR [D]ISK?"
210 DO:GET KE$:LOOP UNTIL INSTR(
"TD",KE$)
220 IF KE$="T"THEN POKE 208,1:EL
SE POKE 208,8
230 PRINT"▣INSERT PRINTFX ";:IF
KE$="T"THEN PRINT"TAPE";:ELSE PR
INT"DISK";
240 PRINT" AND PRESS RETURN"
250 DO:GET KE$:LOOP UNTIL KE$=CH
R$(13)
260 FOR BY=0 TO 3:POKE 209+BY,PE
EK(43+BY):NEXT BY
270 PRINT"▣SAVING PRINTFX BASIC
LOADER..."
280 SAVE"PRINTFX LOADER",PEEK(20
8)
290 PRINT"▣SAVING PRINTFX SYSTEM
 PROGRAM..."
300 POKE 43,1:POKE 44,16:POKE 45
,253:POKE 46,23
310 SAVE"PRINTFX",PEEK(208)
320 POKE 43,PEEK(209):POKE 44,PE
EK(210):POKE 45,PEEK(211):POKE 4
6,PEEK(212)
330 PRINT"▣SAVING COMPLETE - RES
ET MACHINE"
340 END
910 REM
920 REM **************************
*****
930 REM *
 *
940 REM * MACHINE-CODE DATA SECT
ION *
950 REM *
 *
960 REM *    (START AT LINE 1000
)    *
970 REM *
 *
980 REM **************************
*****
990 REM
```

```
10 REM **************************
****
20 REM *
 *
30 REM *    PRINTFX SCREEN EDITOR
 
40 REM *
 *
50 REM * (LOAD & RUN PRINTFX FIR
ST *
60 REM *
 *
70 REM **************************
****
80 REM
90 COLOR 4,1,0:COLOR 0,1,0:COLOR
 1,2
100 GRAPHIC 1,1
110 STANDARD:HEIGHT 1
120 COLOR 1,7,5:TEXT 7,9,"THE PR
INTFX SCREEN EDITOR"
130 COLOR 1,8,5:TEXT 3,12,"(C) C
OPYRIGHT 1988 M.R EVERINGHAM"
140 COLOR 1,2:BOX 1,0,64,319,117
:VOL 8
150 KEY 1,"▣":KEY 2,"▣":KEY 3,"▣
":KEY 4,"▣":KEY 5,"▣":KEY 6,"▣"
160 KEY 7,"▣MY NAME IS SOCRATES▣
"+CHR$(13)
170 KEY 8,"▣WILL HIS EGO KNOW NO
 BOUNDS?▣"+CHR$(13)
180 FOR T=1 TO 200:IF T/50=INT(T
/50) THEN SOUND 1,800,3
190 NEXT T:SCNCLR
200 X=0:Y=0:E=0:W=1:H=1:T=0:U=-1
210 EFFECT E:MODE INVERT:REVERSE
 ON:TEXT X,Y," "
220 DO:GET K$:LOOP UNTIL K$:K=AS
C(K$)
230 IF K<32 OR INSTR("▣▣▣▣▣",K$)
 THEN 270
240 EFFECT E:MODE STORE:TURN T:T
EXT X,Y,K$
250 IF (X<>40-W OR Y<>25-H) THEN
 X=X+W:IF X+W>40 THEN X=0:Y=Y+H
260 GOTO 210
270 TEXT X,Y," ":EFFECT E:MODE S
TORE
280 IF K=13 AND Y+H<=25-H THEN X
=0:Y=Y+H
290 IF K=20 THEN 460
300 IF K$="▣" THEN E=(E AND 127)
 OR 128-(E AND 128)
310 IF K$="▣" THEN E=(E AND 191)
 OR 64-(E AND 64)
320 IF K$="▣" THEN E=(E AND 223)
 OR 32-(E AND 32)
330 IF K$="▣" THEN E=(E AND 239)
 OR 16-(E AND 16)
340 IF K$="▣" THEN E=(E AND 247)
 OR 8-(E AND 8)
350 IF K$="▣" THEN E=(E AND 251)
 OR 4-(E AND 4)
360 IF K$="▣" AND X<>39 THEN E=(
E AND 253)OR 2-(E AND 2):W=3-W
370 IF K$="▣" AND Y<>24 THEN E=(
E AND 254)OR 1-(E AND 1):H=3-H
380 IF K$="▣" THEN T=(T+1) AND 3
390 IF K$="▣" THEN U=-U:IF U<0 T
HEN UPPERCASE:ELSE IF K$="▣" THE
N LOWERCASE
400 IF K$="▣" THEN E=0:U=-1:W=1:
H=1:T=0:UPPERCASE
410 IF K$="▣" OR K$="▣" THEN X=0
:Y=0:IF K$="▣" THEN SCNCLR
420 X=X-W*(K$="▣" AND X+W<=40-W)
+W*(K$="▣" AND X>=W)
430 Y=Y-H*(K$="▣" AND Y+H<=25-H)
+H*(K$="▣" AND Y=>H)
440 IF K=27 THEN 490
450 GOTO 210
460 TEXT X,Y," "
470 IF X<>0 OR Y<>0 THEN X=X-W:I
F X<0 THEN X=40-W:Y=Y-H
480 TEXT X,Y," ":GOTO 210
490 GRAPHIC 0:PRINT CHR$(27)"NPR
INTFX SCREEN EDITOR ABORTED"
500 END
```

Dos 6.1

We put everything you have learned in this series to some practicle use

By Fergal Moane

This final article of the series presents a complete disk operating system, which puts into practice some of the theory already learned. If you are interested, disassembly of the code from $C26B onwards should be useful.

Dos 6.1 - The Utility

This utility is in the same mould as DOS 5.1 on the demo disk. DOS 5.1 is the most used Commodore utility in America. This system outstrips it in a number of ways:

1) DOS 6.1 actually tokenises it's new commands. Tokens are one byte values representing commands, and are used by ordinary Basic for all commands. This ensures compatibility, space economy, and no need for a prefixing character.
2) The above feature means that DOS 6.1 commands can be used in program mode, unlike most normal extensions.
3) DOS 6.1 provides more comprehensive commands in an easily understood form.
4) You are left around 3K free in the 4K block of memory from $C000-$D000
5) Loading is under ten seconds, providing you save out the code seperately, and the program is easily copied to your own disks.

Users Guide

DISK ERROR STATUS
DERR displays the status of the disk drive at the current position on the screen. See your manual for a full explanation of error messages.
Syntax: DERR

DOSS COMMAND
DOS sends a command to the disk drive, opening up a range of around 50 disk commands. Again, see a manual for details of standard disk commands. If DOS is used without a command, the error status will be returned.
Syntax: DOS"command"

DIRECTORY
Displays the directory of drive 0 or 1 (Usually drive 0) without loading it into memory.
Syntax: DIR, drive number

MEMORY SAVE
Saves memory from start address to end address to the specified device. This is very useful with graphics or machine code data.
Syntax: MSAVE, start, end, "filename", dev

DEFAULT TO DISK
This sets device 8 as the default device for all SAVE and LOAD operations. LOAD"filename will now have the same effect as LOAD"filename",8
Syntax: DISK

DEFAULT TO TAPE
Tape is now the default device for SAVing and LOAding.
Syntax: TAPE

NORMAL DEVICE NUMBERS
This returns to the normal Commodore device numbering system. Use this if you need to load with ,8,1
Syntax: NORMAL

LIST BASIC PROGRAM
DLIST allows you to list a Basic program directly off disk without loading it into memory. This is useful in checking a program, or grabbing lines from another program. RUN/STOP will stop the listing at any time. If you try to list a non-basic program, you will get the expected garbage. Press RUN/STOP RESTORE followed by CLOSE2 to recover.
Syntax: DLIST,"filename"

LEAVE DOS 6.1
Leaves the DOS 6.1 system without disturbing Basic or any memory. SYS49152: SYS49676 will return you to the DOS 6.1 system.
Syntax: QUIT

MEMORY USAGE
A memory map of the system could be useful:

C000 – C120	Program for adding new tokens
C150 – C1B1	Table of jump addresses for new commands
C2000 – C53A	New Dos routines
D53B – CFFF	Free RAM for user programs

Some low memory and zero page locations are also altered during the command processing, but this has no effect on Basic and the area around the tape buffer is untouched.

Loading DOS 6.1

Type in and save the Basic program. When run, you will be presented wth a screen of instructions. The machine code is being POKed into memory at this stage. On completion of the program NEWs itself and you are into the DOS system. Obviously, you will not want to keep having to load the basic loader each time you wish to use it. Therefore, I suggest you use the MSAVE command to save out the relevant two portions of memory.

USING DOS 6.1 IN YOUR OWN PROGRAM
As the commands can be used in your own programs, it would be inconvenient to have to load DOS 6.1 in every time, expecially if your program is seen by others. Follow the following procedure to include it in your program.

Load the two machine code files that you have saved at the start of your program. Immediately afterwards SYS49152: SYS49676 to initialise the commands.

Program Notes

When entering programs in the modified environment, C64 Basic insists that you now have to enter colons after a THEN statement. There seems to be no way to overide this, but it does not effect programs saved in the normal envi These will work normally.

EG:- 10 GET A8: IF A$= "" THEN: GOTO 10

20 IFA$="!" THEN: DERR

Also note that some commands will return to direct mode if used in a program. A bit of POKing of the keyboard buffer could solve this. Unfortunately, this is unavoidable.

I hope that this series of articles has been of use to you, and makes your disk drive easier to handle.

```
PROGRAM:DOS 6.1

1D   10 REM ********************
30   11 REM IF YOU WANT TO SAVE
90   12 REM TIME YOU CAN SAVE
AE   13 REM THE MACHINE CODE BY
37   14 REM THE USE OF A MONITOR
38   15 REM PROGRAM.
BF   16 REM ********************
27   21 PRINT"[CLR,GREEN]SETTING
     UP DOS 6.1.....PLEASE WAIT":
     POKE53281,0:POKE53280,0
1E   22 PRINT"[DOWN,YELLOW]COPYRI
     GHT 1989[SPC3]SENSELESS SOFT
     WARE[GREEN]"
39   23 PRINT"[DOWN]THE NEW COMMA
     NDS ARE:":PRINT"[S*21]"
84   24 PRINT"DERR : RETURNS DISK
      ERROR STATUS"
5C   25 PRINT"[DOWN]DOS : ALLOWS
     COMMANDS TO THE DISK DRIVE"
14   26 PRINT"[DOWN]DIR,0 : DISPL
     AYS THE DIRECTORY"
22   27 PRINT"[DOWN]MSAVE : SAVES
      ANY BLOCK OF MEMORY"
E3   28 PRINT"[DOWN]DISK : SETS U
     P THE DISK AS THE DEFUALT"
17   29 PRINT"[DOWN]TAPE : SETS U
     P TAPE AS THE DEFAULT"
7A   30 PRINT"[DOWN]NORMAL : USES
      NORMAL DEVICE NUMBERS"
E7   31 PRINT"[DOWN]DLIST,NAME :
     LISTS ANY PROGRAM WITHOUT  L
     OADING IT INTO MEMORY"
F1   32 PRINT"[DOWN]QUIT : EXITS
     DOS 6.1"
19   33 GOTO670
05   50 PRINT"[CLR,DOWN,RIGHT4]**
     * DISK OPERATING SYSTEM 6.1
     ***"
74   60 PRINT"[DOWN,RIGHT4]COPYRI
     GHT SENSELESS SOFTWARE 1989"
DF   70 PRINT"[DOWN] 64K RAM SYST
     EM  38911 BASIC BYTES FREE"
6B   80 SYS49152:SYS49676:NEW
DD   670 I=679
0A   671 READ A:IF A=256 THEN 490
     00
DF   672 POKE I,A:I=I+1:GOTO 671
86   679 DATA 169,8,141,230,2,141
     ,241,2
E0   687 DATA 169,229,141,48,3,16
     9,240,141
CA   695 DATA 50,3,169,2,141,49,3
     ,169
FC   703 DATA 2,141,51,3,96,32,17
     5,2
3D   711 DATA 169,1,141,230,2,141
     ,241,2
6C   719 DATA 96,169,165,141,48,3
     ,169,244
E6   727 DATA 141,49,3,169,237,14
     1,50,3
7B   735 DATA 169,245,141,51,3,96
     ,169,8
49   743 DATA 133,186,169,0,133,1
     0,76,165
C0   751 DATA 244,169,8,133,186,7
     6,237,245
2F   759 DATA 0,256
83   49000 I=49152
66   49020 READ A:IF A=256 THEN 5
     0
E5   49030 POKE I,A:I=I+1:GOTO 49
     020
6D   49152 DATA 162,21,160,192,14
     2,4,3,140
86   49160 DATA 5,3,162,219,160,1
     92,142,6
37   49168 DATA 3,140,7,3,96,166,
     122,160
27   49176 DATA 4,132,15,189,0,2,
     16,10
ED   49184 DATA 197,255,240,65,23
     2,208,62,232
AD   49192 DATA 208,241,201,32,24
     0,55,133,8
62   49200 DATA 201,34,240,85,36,
     15,112,45
F6   49208 DATA 201,63,208,4,169,
     153,208,37
06   49216 DATA 201,48,144,4,201,
     60,144,29
B9   49224 DATA 132,113,160,0,132
     ,11,136,134
EE   49232 DATA 122,202,200,232,1
     89,0,2,56
1E   49240 DATA 249,158,160,240,2
     45,201,128,208
F5   49248 DATA 47,5,11,164,113,2
     32,200,153
B7   49256 DATA 251,1,201,0,240,5
     6,56,233
54   49264 DATA 58,240,4,201,73,2
     08,2,133
15   49272 DATA 15,56,233,85,208,
     157,133,8
96   49280 DATA 189,0,2,240,224,1
     97,8,240
24   49288 DATA 220,200,153,251,1
     ,232,208,240
38   49296 DATA 166,122,230,11,20
     0,185,157,160
B8   49304 DATA 16,250,185,158,16
     0,208,181,240
41   49312 DATA 15,189,0,2,16,189
     ,153,253
A5   49320 DATA 1,198,123,169,255
     ,133,122,96
F2   49328 DATA 160,0,185,16,193,
     208,2,200
26   49336 DATA 232,189,0,2,56,24
     9,16,193
61   49344 DATA 240,245,201,128,2
     08,4,5,11
43   49352 DATA 208,153,166,122,2
     30,11,200,185
A4   49360 DATA 15,193,16,250,185
     ,16,193,208
5D   49368 DATA 224,240,198,16,15
     ,36,15,48
58   49376 DATA 11,201,255,240,7,
     201,204,176
```


```
5D   49368 DATA 224,240,198,16,15
     ,36,15,48
58   49376 DATA 11,201,255,240,7,
     201,204,176
```


```
E5   49384 DATA 6,76,36,167,76,24
     3,166,56
A7   49392 DATA 233,203,170,132,7
     3,160,255,202
A1   49400 DATA 240,8,200,185,16,
     193,16,250
72   49408 DATA 48,245,200,185,16
     ,193,48,5
93   49416 DATA 32,71,171,208,245
     ,76,239,166
2A   49424 DATA 68,69,82,210,68,7
     9,211,68
D4   49432 DATA 73,210,77,83,65,8
     6,197,68
C5   49440 DATA 73,83,203,84,65,8
     0,197,78
F4   49448 DATA 79,82,77,65,204,6
     8,76,73
15   49456 DATA 83,212,81,85,73,2
     12,3,140
8C   49464 DATA 9,3,169,108,160,1
     93,141,10
FC   49472 DATA 3,140,11,3,96,32,
     115,0
2F   49480 DATA 32,78,193,76,174,
     167,201,204
A8   49488 DATA 144,4,201,251,144
     ,6,32,121
6B   49496 DATA 0,76,237,167,56,2
     33,204,10
60   49504 DATA 170,189,159,193,7
     2,189,158,193
CC   49512 DATA 72,76,115,0,169,0
     ,133,13
C7   49520 DATA 32,115,0,201,251,
     144,4,201
DB   49528 DATA 255,144,6,32,121,
     0,76,141
B3   49536 DATA 174,56,233,251,10
     ,72,32,115
ED   49544 DATA 0,32,241,174,104,
     170,185,172
B4   49552 DATA 193,133,85,185,17
     3,193,133,86
67   49560 DATA 32,84,0,76,141,17
     3,107,194
02   49568 DATA 107,194,25,195,24
     0,195,166,2
B7   49576 DATA 195,2,207,2,226,2
     52,0,0
30   49584 DATA 0,165,1,41,254,13
     3,1,177
86   49592 DATA 87,221,128,191,8,
     165,1,9
E1   49600 DATA 1,133,1,40,208,7,
     200,232
2B   49608 DATA 228,34,208,229,96
     ,230,35,164
89   49616 DATA 35,162,0,177,87,2
     40,3,76
19   49624 DATA 177,193,165,87,56
     ,233,2,133
0F   49632 DATA 87,165,88,233,0,1
     33,88,160
40   49640 DATA 0,177,87,133,89,2
     00,177,87
F7   49648 DATA 133,88,5,89,240,1
     6,165,89
3A   49656 DATA 24,105,2,133,87,1
     65,88,105
A7   49664 DATA 0,133,88,76,171,1
     93,104,104
87   49672 DATA 32,93,194,120,169
     ,23,160,194
B7   49680 DATA 141,8,3,140,9,3,9
     6,32
97   49688 DATA 115,0,32,32,194,7
     6,174,167
CF   49696 DATA 201,204,144,4,201
     ,251,144,6
66   49704 DATA 32,121,0,76,237,1
     67,56,233
B8   49712 DATA 204,10,170,189,63
     ,194,72,189
F0   49720 DATA 62,194,72,76,115,
     0,107,194
```

Dos 6.1

```
13  49728 DATA 107,194,25,195,24
    0,195,166,2
69  49736 DATA 195,2,207,2,64,19
    6,82,228
05  49744 DATA 87,173,107,194,13
    3,88,230,35
FB  49752 DATA 164,35,162,0,96,2
    30,122,208
84  49760 DATA 2,230,123,160,0,1
    77,122,96
42  49768 DATA 0,0,0,0,32,121,0,
    240
8A  49776 DATA 3,76,226,194,169,
    0,133,183
EB  49784 DATA 32,176,194,169,13
    ,32,210,255
EA  49792 DATA 169,18,32,210,255
    ,162,15,32
5F  49800 DATA 198,255,32,207,25
    5,72,165,144
66  49808 DATA 208,7,104,32,210,
    255,76,138
D4  49816 DATA 194,104,169,15,13
    3,73,32,204
C5  49824 DATA 225,169,146,32,21
    0,255,169,13
C2  49832 DATA 32,210,255,169,0,
    76,198,255
81  49840 DATA 32,201,194,141,15
    5,194,141,234
11  49848 DATA 194,141,134,194,1
    33,184,169,15
E4  49856 DATA 133,185,169,8,133
    ,186,76,193
07  49864 DATA 225,169,15,166,15
    2,224,0,240
B2  49872 DATA 16,221,88,2,208,6
    ,56,233
90  49880 DATA 1,76,203,194,202,
    240,2,208
F6  49888 DATA 240,96,201,34,240
    ,3,76,8
53  49896 DATA 175,169,15,133,73
    ,32,204,225
EB  49904 DATA 32,115,0,165,122,
    133,187,165
E3  49912 DATA 123,133,188,162,0
    ,32,99,194
8B  49920 DATA 240,11,201,34,240
    ,7,232,32
BB  49928 DATA 93,194,76,0,195,1
    34,183,32
D1  49936 DATA 176,194,169,13,32
    ,210,255,76
D5  49944 DATA 116,164,32,253,17
    4,32,121,0
3E  49952 DATA 240,11,201,48,240
    ,7,201,49
25  49960 DATA 240,7,76,8,175,16
    9,48,208
AC  49968 DATA 2,169,49,141,235,
    195,169,2
F1  49976 DATA 162,234,160,195,3
    2,189,255,169
41  49984 DATA 14,32,203,194,162
    ,8,160,0
9F  49992 DATA 32,186,255,32,192
    ,255,144,10
68  50000 DATA 72,165,184,32,195
    ,255,104,76
F5  50008 DATA 249,224,160,3,132
    ,183,166,184
E9  50016 DATA 32,198,255,32,207
    ,255,133,87
A9  50024 DATA 32,183,255,208,11
    4,32,207,255
CC  50032 DATA 133,88,32,183,255
    ,208,104,164
08  50040 DATA 183,136,208,224,1
    32,183,32,207
59  50048 DATA 255,72,32,183,255
    ,170,104,224
41  50056 DATA 0,208,84,164,183,
    192,80,176
47  50064 DATA 78,153,0,2,170,24
    0,4,230
AC  50072 DATA 183,208,227,32,20
    4,255,166,159
0F  50080 DATA 224,3,240,5,166,1
    58,32,201
A4  50088 DATA 255,166,87,165,88
    ,32,205,189
F0  50096 DATA 169,32,32,210,255
    ,160,0,185
EE  50104 DATA 0,2,240,6,32,210,
    255,200
98  50112 DATA 208,245,169,13,32
    ,210,255,32
4A  50120 DATA 204,255,32,225,25
    5,240,16,32
68  50128 DATA 228,255,201,32,20
    8,5,32,228
39  50136 DATA 255,240,251,160,2
    ,208,155,32
4A  50144 DATA 204,255,165,184,3
    2,195,255,76
9E  50152 DATA 123,227,36,48,0,0
    ,0,0
DF  50160 DATA 0,165,43,72,165,4
    4,72,165
B2  50168 DATA 45,72,165,46,72,3
    2,121,0
```


```
DF  50160 DATA 0,165,43,72,165,4
    4,72,165
B2  50168 DATA 45,72,165,46,72,3
    2,121,0
```


```
92  50176 DATA 32,253,174,32,138
    ,173,32,247
7E  50184 DATA 183,165,20,133,25
    1,165,21,133
8C  50192 DATA 252,32,253,174,32
    ,138,173,32
8E  50200 DATA 247,183,165,20,13
    3,253,165,21
43  50208 DATA 133,254,32,253,17
    4,32,212,225
C6  50216 DATA 169,251,166,253,1
    64,254,32,216
A9  50224 DATA 255,104,133,46,10
    4,133,45,104
5A  50232 DATA 133,44,104,133,43
    ,96,0,0
1E  50240 DATA 0,32,115,0,240,5,
    32,87
F2  50248 DATA 226,176,1,96,173,
    255,159,201
D5  50256 DATA 36,208,3,76,4,193
    ,169,2
4A  50264 DATA 162,8,160,0,32,18
    6,255,32
B0  50272 DATA 192,255,169,0,32,
    189,255,169
A9  50280 DATA 15,162,8,160,15,3
    2,186,255
22  50288 DATA 32,192,255,162,15
    ,32,198,255
1D  50296 DATA 160,2,32,207,255,
    201,48,208
E7  50304 DATA 5,136,208,246,240
    ,32,32,207
BC  50312 DATA 255,32,207,255,32
    ,207,255,201
A0  50320 DATA 44,240,5,32,210,2
    55,208,244
0C  50328 DATA 169,15,32,195,255
    ,169,2,32
CB  50336 DATA 195,255,32,204,25
    5,96,162,2
F2  50344 DATA 32,198,255,32,207
    ,255,32,207
4A  50352 DATA 255,165,145,201,1
    27,240,225,201
4F  50360 DATA 239,240,246,32,20
    7,255,32,207
3F  50368 DATA 255,240,213,32,20
    7,255,72,32
DE  50376 DATA 207,255,168,104,1
    70,152,32,205
44  50384 DATA 189,169,32,32,210
    ,255,32,207
E6  50392 DATA 255,208,7,169,13,
    32,210,255
2D  50400 DATA 208,207,201,34,20
    8,16,32,50
AF  50408 DATA 197,32,207,255,20
    1,34,240,18
EB  50416 DATA 201,13,240,14,208
    ,240,133,252
A1  50424 DATA 165,212,208,6,165
    ,252,201,128
27  50432 DATA 176,5,32,50,197,1
    44,207,41
```

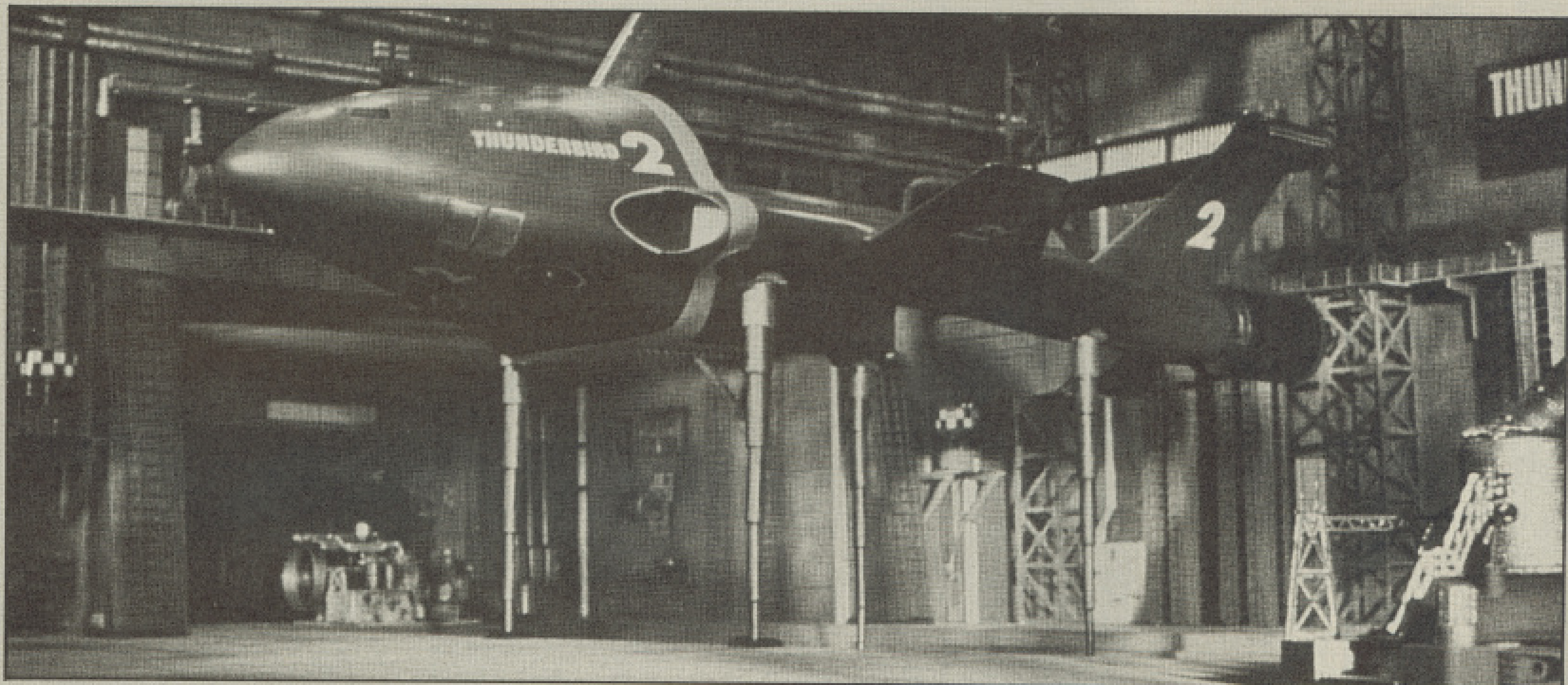

```
A1  50424 DATA 165,212,208,6,165
    ,252,201,128
27  50432 DATA 176,5,32,50,197,1
    44,207,41
```


```
C2  50440 DATA 127,133,252,230,2
    52,162,0,160
D4  50448 DATA 0,185,157,160,201
    ,128,144,1
19  50456 DATA 232,200,228,252,2
    08,243,240,4
CC  50464 DATA 32,50,197,200,185
    ,157,160,201
45  50472 DATA 128,144,245,41,12
    7,32,50,197
36  50480 DATA 144,164,201,32,14
    4,3,32,210
DA  50488 DATA 255,96,0,256
```

COMPETITION

ENTER

No Strings Attached



Win wonderful prizes, care of Grandslam, in our easy-to-enter Thunderbirds Competition.

Five first prize winners – drawn after the closing date – will receive a Thunderbirds video and the game for Amiga or C64. Pens and posters are available for ten runners-up.

All you have to do is write the correct answers to the following questions and send them, on the back of a postcard or sealed envelope, to:

Thunderbirds Compo, Your Commodore, Argus House, Boundary Way, Hemel Hempstead, HP2 7ST.

1. What is Brain's real name?
a) Horatio Hackenbacker
b) Bartholomew Baker
c) Cuthbert Cooke

2. What does FAB stand for?
a) Futuristic Automatic Breakthrough
b) Nothing
c) Funderbirds Are Brill

3. What is Parker's nickname?
a) Nosey
b) Shifty
c) Buster

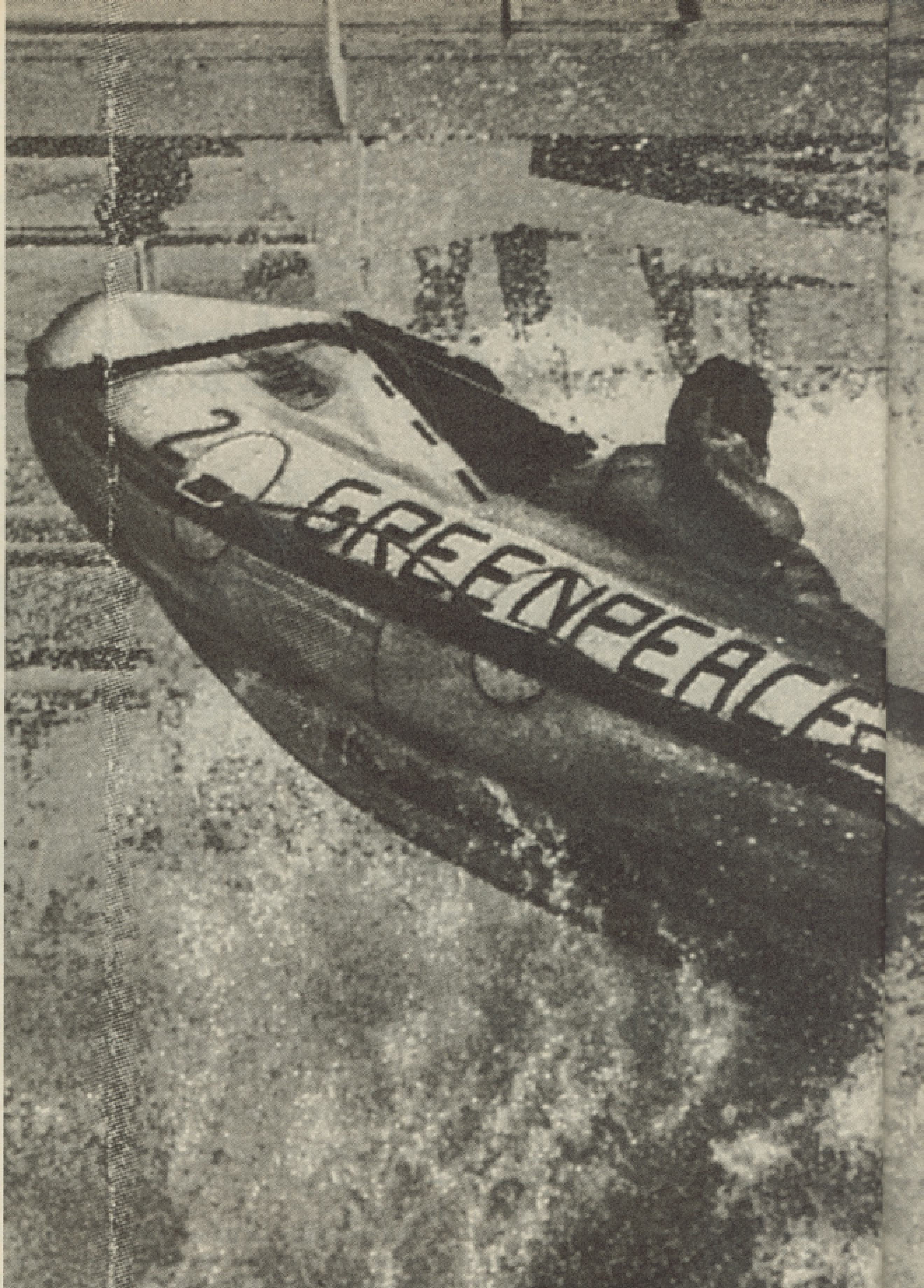
Closing date: 31st August 1989.

The Rules

Entries will not be accepted from employees of Argus Specialist Publications, Grandslam, or Teque. This restriction also applies to employees families and agents of the companies.

The entry restrictions form part of the rules and the Editor's decision is final. No correspondence will be entered into. In the event of a postal strike, we reserve the right to extend the closing date.

This Green and Pleasant Land!



Emma Norman dons her green outfit and, with banner grasped firmly in hand, campaigns for the organisation that everybody should know about

In a recent advertising campaign the organisation Friends Of the Earth gave five billion reasons why you should join them: The Human Race. These same five billion reasons could be given for joining the environmental organisation Greenpeace. Both Greenpeace and Friends of the Earth have the same object at heart: To conserve the planet for future generations.

Most people have heard of Greenpeace and are aware of the work they do, but for those who do not know it is, in its own words: "An International Environmental Pressure group which maintains complete independence from all political parties anywhere



in the world. Its aim is to bring an end to the pollution of the planet and to protect wildlife." It operates throughout the world.

It began in 1971 in Canada, with a protest voyage into a nuclear test zone at Amchitka. The test was disrupted and the area is now a bird sanctuary. In 1977 Greenpeace opened its first branch in Britain where, to date, it has over 250,000 members. Today there are over 3½ million members Worldwide with the majority of these coming from the USA. This figure is one of which they are very proud, but more members are always needed. Despite the successes, there are many objectives still to be reached.

Why do we need Organisations like Greenpeace? What is so harmful with dumping waste? The wastes are concoctions of poisonous substances: metals like Copper, Lead, Zinc, Chlorine, Mercury and Arsenic. These build up in the environment and poison food sources. For example: In Wood Spring, North Devon people were warned not to eat more than 1lb of Shrimps or 4oz of shellfish due to the excess of cadmium (a harmful poison) therein. Two million tonnes per year of such waste is dumped in the Irish sea.

Although a direct connection has not been established between waste-dumping and the virus that killed 17,000 seals last year (and appears to be re-emerging amongst the remaining Common Seal population), the possibility that pollution was the cause cannot be overlooked and emphasises the urgency for action. There are many more examples of such atrocities caused by "Man's inhumanity to man", including Chernobyl, the Alaska oil spillage, the plight of the whales, elephants; too numerous to list, but what of the successes?

In the past few years Greenpeace has: Stopped hunts of seal pups; helped bring about an end in commercial whaling; stopped the burning of hazardous waste in the North Sea, US waters and the Mediterranean; stopped the dumping of radioactive wastes at sea; helped persuade the British Government to spend millions of pounds cleaning the beaches in Britain and filtering the waste from coal-fired power stations.

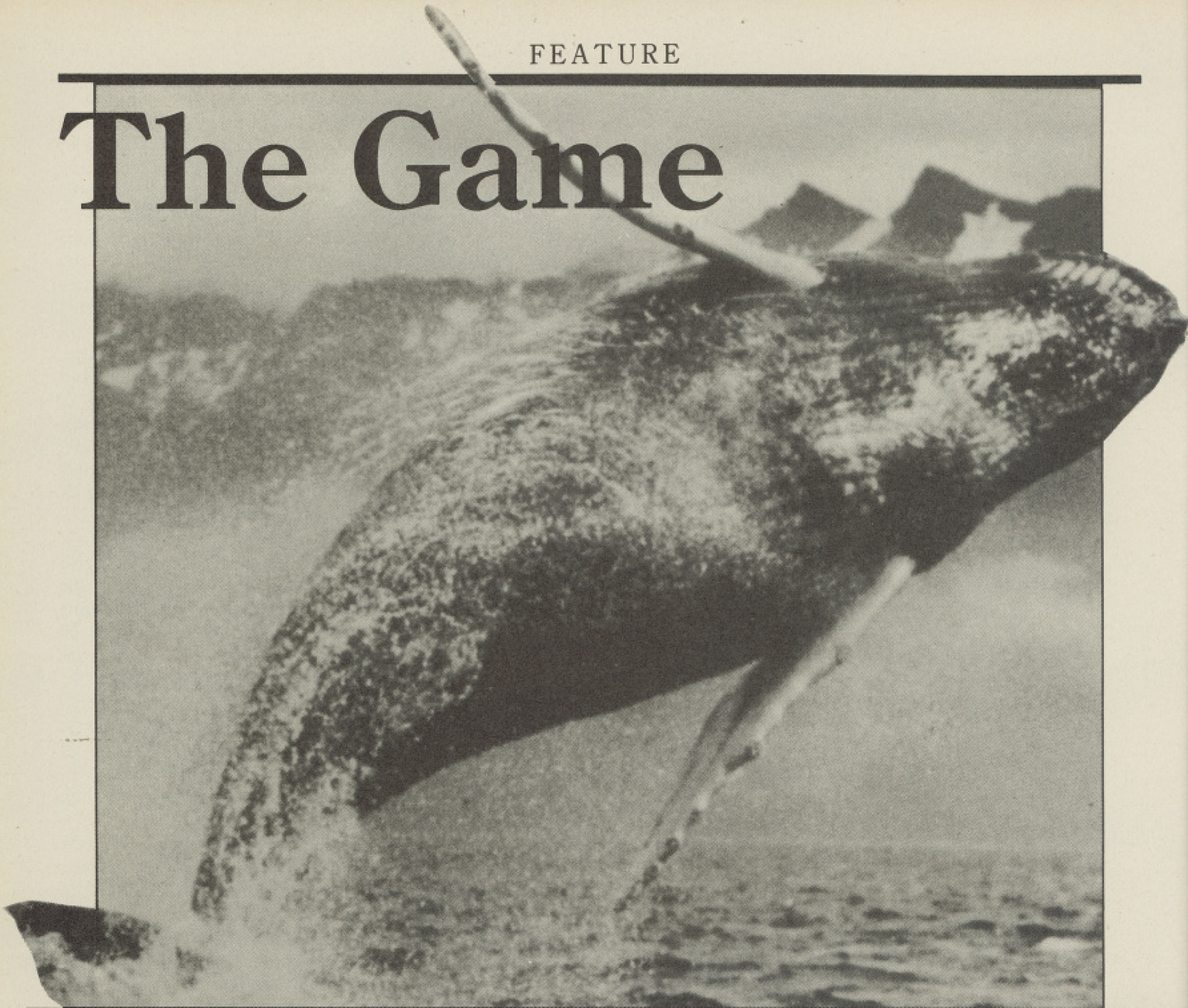
The majority of these achievements are brought about by sheer bravery. Greenpeace activists went out in dinghyies and positioned themselves between the harpoons of the whaling boats and the fleeing whales. As a result Commercial whaling is now banned. In the North Atlantic, Greenpeace drove its inflatable dinghyies under barrels of radioactive waste, therefore dumping of such waste at sea has now been stopped. Volunteers physically prevented baby seals being killed in the Orkney Isles and Newfoundland by throwing themselves between the hunters and the pups.

The level of the danger in which these people place themselves became apparent when one man was killed on the Rainbow Warrior when it was destroyed whilst trying to stop the testing of nuclear weapons off the coast of New Zealand. These people are risking their lives? What can you do to help?

You can join them in their fight to protect the planet by becoming a member, information and membership forms are available. from: *Greenpeace 30-31 Islington Green, London N1 8BR.* (Subscriptions cost £12 for a single, £17.50 for the family, £6 unwaged, and £20 for overseas).

***Greenpeace** is at present campaigning for Nuclear free seas in Germany, Nuclear free Irish sea, stopping toxic waste dumping in Spain, working to protect the seals and dolphins round the British coast, and trying to get Antarctica declared a World Park.*

The Game

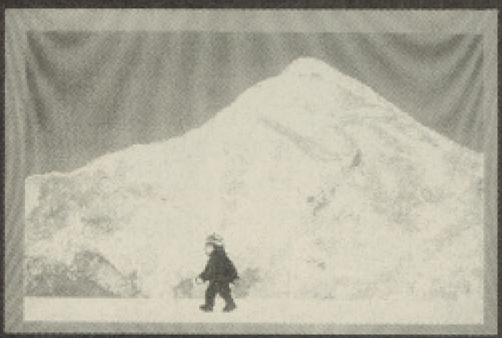


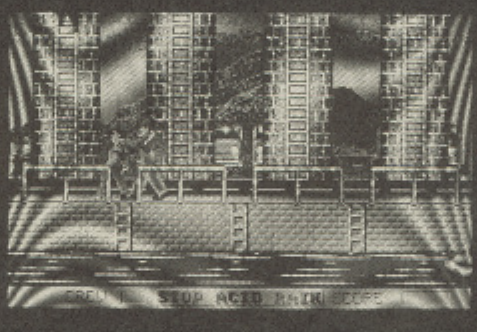
ENTITLED 'Rainbow Warrior', the Greenpeace game is set to raise a few eyebrows. It has already provoked a Conservative MP to condemn it, believing that many computer gamers may set about industrial pipelines and use other forms of violence in order to 'practise' the Greenpeace methods shown in the game.

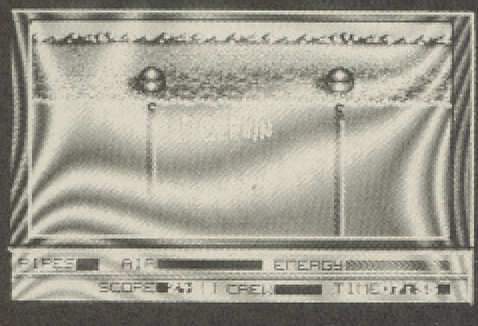
The gameplay is split into several sections, all of which are based on separate ecological problems. You play a Greenpeace supporter who is determined to protect wildlife and stop pollution of the natural world, and each section is a mini-game all in itself. Among other tasks you have to prevent whaling ships from catching their prey by manoeuvring the Greenpeace boat between the whale and the ship, save penguins from straying into a beam that has broken through the ozone layer, and you must stop seal culling by ruining the pelts of cubs, this is done by spraying them with green paint.

Although the game features very strong opinions, Microprose has made the graphics slightly humorous in tone, so as to appeal to a wider market. The sprites are enormous and the animation as smooth as baby's bottom, they have even been described as Python-esque.

Rainbow Warrior should now be available on Amiga and C64, and it is to coincide with the new Greenpeace album, Rainbow Warriers, which features such artists as U2 and Simple Minds.







RAINBOW WARRIOR

RAINBOW WARRIORS

The worlds first environment friendly software!

At last, an all action game that presents a solution to the environmental dilemma faced by the entire human race. Rainbow Warriors is an action game with real depth and meaning. It simulates seven campaigns of Greenpeace members over the last few years.





MicrO-Style

GAMES FOR ADULTS

UNIT 1, HAMPTON ROAD INDUSTRIAL ESTATE, TETBURY, GLOS. TEL: 0666 504326/504412.



THUNDERBIRDS

THERE have been many cult shows from television. By 'cult' I mean shows which have seen more repeats than a particularly bad curry. Examples are The Wooden Tops, Bill and Ben – The Flowerpot men, Andy Pandy and Thunderbirds. All of these have something rather significant in common, their actors were more wooden than the cast of Neighbours, and have more strings than all of Hitchcock's finest thrillers put together. They all featured puppets.

The latter of these reputable shows is the one which we are to spotlight here, mainly due to the fact that International Rescue see their first pixelised exploits, thanks to software company *Grandslam,* whose previous credits include *Pacland, Pacmania* and *The Running Man,* as well as the forthcoming *Liverpool – The Computer Football Game* (which has been widely acclaimed). Thunderbirds is its first venture into the world of Gerry Anderson, although contacts have it that it will not be the last (look out Mysterons...!).

Released in the early sixties, Thunderbirds gained popularity at an alarming rate, due mainly to Anderson's previous successes – *Captain Scarlet, Fireball XL5*, and *Stingray* to name a few – but, there was something rather special about the Tracy family that enthralled viewers everywhere. and the plots were far more detailed than the simple rescue missions that they seemed to portray.

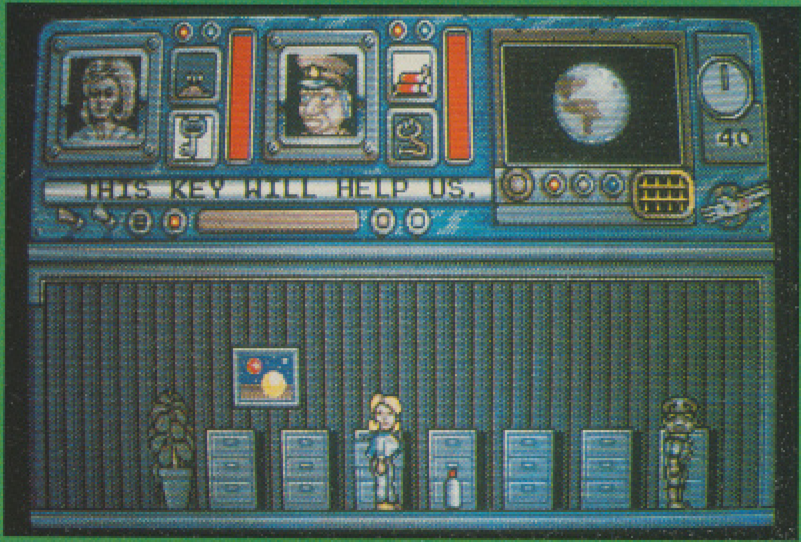
The characters created were not as shallow as one may expect. Each one had a more complex background and history than many at the forefront of soap operas, thus making for a slightly more involving program. The personnel of of International Rescue runs as follows:

Jeff Tracy – Founder of International Rescue, and father of five sons (who are also in the 'family business').

Scott Tracy – Eldest of the brothers, pilot of Thunderbird 1, and second in command if anything happens to his father.

Virgil Tracy – Pilot of Thunderbird 2, also deputises on Thunderbird 3 when Alan is not available.

Alan Tracy – Pilot of Thunderbird 3, also alternates with John on tours of duty on Thunderbird 5. Is romantically involved with TinTin.



Gordon Tracy – Pilot of Thunderbird 4.

John Tracy – Spends most of his time on Thunderbird 5, due to his days as a radio ham. Occasionally replaces Alan on Thunderbird 3.

Brains – Designer of the Thunderbird equipment and technical genius. Has a tendency to stutter.

Kyrano – The Tracy Island cook. Is half-brother of the villainous Hood.

TinTin – Kyrano's daughter is in love with Alan, and helps Brains maintain and repair the Thunderbird equipment.

Lady Penelope – Ex-British agent, famous for her unique pink Rolls Royce FAB 1.

Parker – Failed safe-cracker who was caught by Lady Penelope. Acts as her butler as 'cover'.

All of these except Lady Penelope and Parker were based on Tracy Island, which wowed audiences with its great technical sequences when the Thunderbirds lifted into the air, most memorable of which was the swimming pool which slid back to allow Thunderbird 1 to soar from it's underground launching pad. The craft themselves were superbly designed, thus encouraging kiddies everywhere to request a home version of Thunderbird 2 – the most ingenious of them all – courtesy of Dinky toys. Thunderbird 2 was popular because of it's ability to store other pieces of equipment – namely Thunderbird 4 – in its stomach.

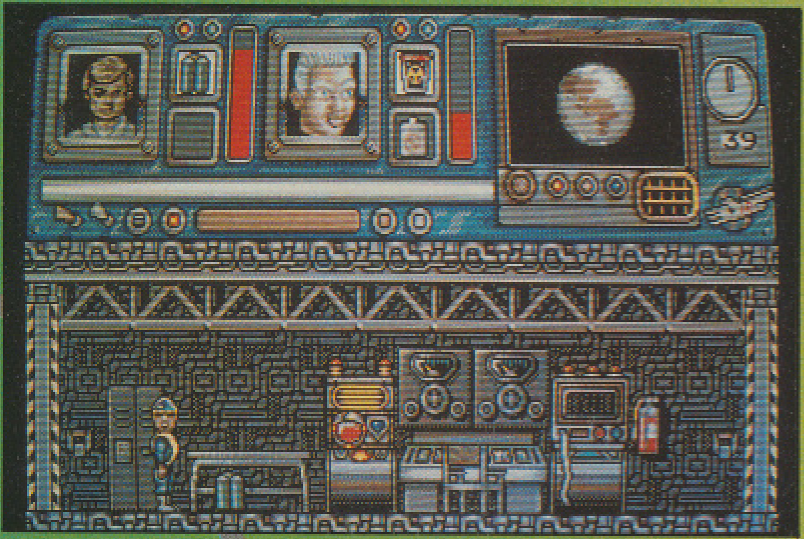
It is this depth, characterisation, gripping plots and technical brilliance that made Thunderbirds the most successful of Gerry Anderson's productions, and it is for this reason that the computer game should not go without notice. The game has many of its own merits, being the best arcade adventure to see

these shores for many months is just the first.

The game features four different missions. You must control two characters in each and perform certain actions in order to come out unscathed (although Thunderbirds can never die, and so sayeth the lord Anderson) and acclaimed as a hero. The first – Mine Menace – involves Alan and Brains who have to rescue some trapped miners. This is not a simple operation as priority lies in shutting off the mine's pumping station before it floods the whole mine, thus rendering all to be rescued by a rival team, causing embarrassment and clients to seek other help – like I said, Thunderbirds cannot die.

The other three levels – Sub Crash, Bank Job and Countdown to Terror respectively – are all in a similar vein, although time limits in which to complete your task get progressively more restrictive, and the problems connected to the mission become harder to figure out.

Graphics throughout are very pretty and true to Anderson's original ideas. Presentation throughout is of a very high quality, for instance, before each mission there is a small digitised animation sequence from the old episodes, for the four sections are all based on existing, single shows, although only loosely so those who own the videos



will find no favours in swotting. The famous theme tune – 5... 4... 3... 2... 1... Thunderbirds are Go!... Bom... Bom... Bom... Bom... etc. – has been reworked to a rather good house version, which accompanies the title sequences but leaves the game itself to sampled sound effects.

Another small thing that I think helps sway my attitude about this game is the small animated sequences which the graphics go through at certain stages (similar to Little Computer People) giving the game more character. Each mission has a password which once discovered allows passage straight into the section without the earlier messy



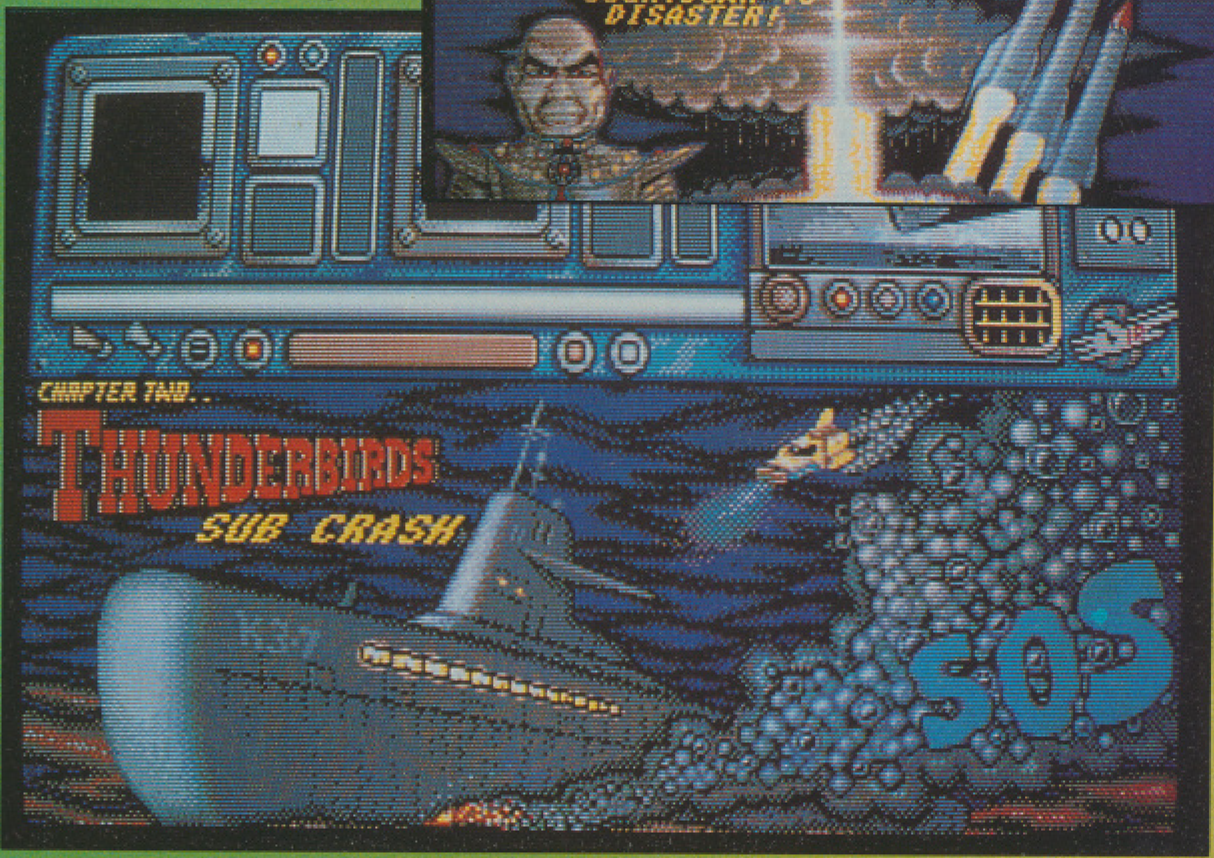
work I think *Teque, Grandslam,* and *Gerry Anderson* will all be very pleased with the final product of some deft licensing and hard work, I know I am as a lifelong Thunderbirds fan and I can see myself getting as gripped by this computer version as I was with the original series. Although it really is one of the oldest clichés in the book this time, for me, Thunderbirds ARE go!!!

Rik Henderson

***Tecque** are the programming team behind Chubby Gristle, Terramex, and the more recent Prison – proving that they are really arcade adventure experts – although Thunderbirds is the most complex of all, and probably the most complete.*

Grandslam – £24.95

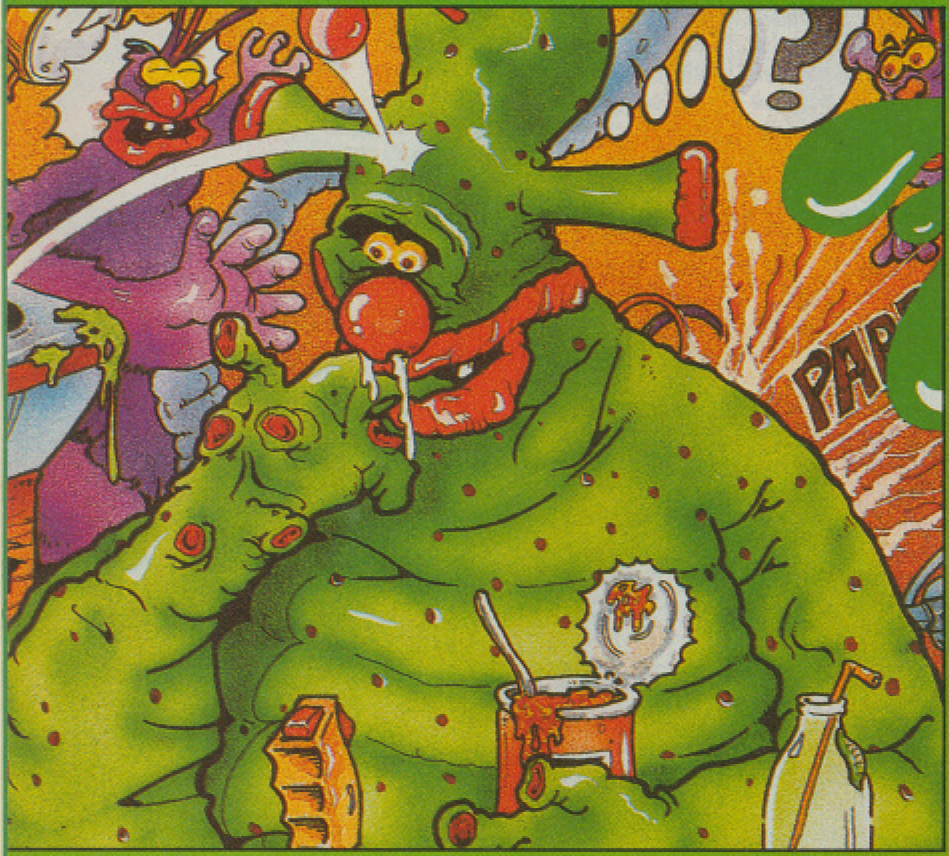




INFO

A

Gameplay: 97%
Graphics: 95%
Sonics: 91%
Lastability: 85%
Overall: 92%



GILBERT - ESCAPE FROM DRILL

Again Again – £9.99/£14.99

Follow Gilbert on his snotty escapades as he explores the planet Drill, whilst searching for the missing parts to the Millenium Dustbin.

THE cute but snotty alien, Gilbert, is hankering once again for the bright lights of ITV. He has the contract, and the ego, now all he needs to do is find the missing parts of the Millenium Dustbin and sign on the dotted line.

Gilbert has twenty four hours to find the five missing parts, hidden by the discerning TV watchers of the planet, before some other, hopefully more talented, star signs the contract. No Drillian possesses a full set of faculties and these would-be saboteurs have actually left clues in various video games, which should help Gilbert fulfill his quest in snottifying the minds of the Earthly young.

Gilbert starts his adventure in the middle of his home town, though why someone would wish to build a town in such a hostile place is beyond me. As you slither – yes Gilbert has no feet – across town you are assaulted by all manner of beasts. Fortunately Gilbert can dispatch these creatures with a flick of the head and a globule of snot. This formidable method of defence is finite but fresh supplies of the sticky green stuff can be found in milk bars, along with the video games which gave the missing parts. If Gilbert manages to shoot an entire wave of creatures, a flying plate of wobbling jelly gyrates its way onto the screen. Shooting this either liberates nothing, a can of beans, or a piece of cake. Of which four of these items can be carried at any one time. If the beans are eaten Gilbert bloats and floats, enabling him to reach the upper levels of the forest and change levels when in the sewer system. Eating the cake brings Gilbert back to Drill with a bump, obviously made with the wrong brand of flour.

The five video games which Gilbert must complete offer a welcome break from the more mundane pleasures of snotting monsters in the street. Some of the better games include Earth Invaders – in which a one eyed monster blasts at rows of descending humans. Brain Drain – play against the computer to see who can match the most tiles. And Snotfight at the OK Corral – snot 'em cowboy. Failure to complete a video game produces a loss of one hour of time.

Unlike other games the attacking creatures never cease to amaze as they come forth in a never ending variety of guises. The playing area is not that small, and not only is there a cityscape to contend with, but a jungle, sewer, tree top and underwater adventures to occupy the mind. Snotting good fun even if the gameplay is a little too repetitive.

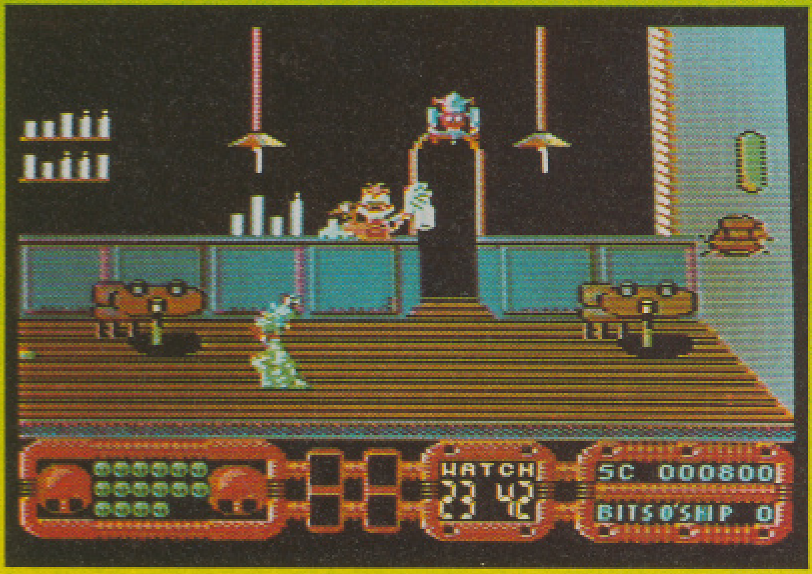
Adrian Pumphrey

INFO

64

Gameplay: 68%
Graphics: 82%
Sonics: 76%
Lastability: 58%
Overall: 70%

Enigma Variations, *headed by Mark Greenshields and Richard Naylor, are the people responsible for the latest TV presenter computer game and as a publicity stunt they sent all the hacks a free sticky Gilbert to throw around – what fun that was.*



SCRAMBLERS, in my opinion, could be so called because a couple of hours on the back of one of these things it's scrambled brains for tea, dear. The only advantage posed here is one of less excitement coupled with even less pain than the real thing.

Fifteen tracks of pure frustration lie before you, and failure to complete one within the time limit means a trip all the way back to the beginning. You start with the three dirt tracks beginning with the ever so easy track A, which is designed to allow you to save face by letting you complete at least one of the fifteen tracks. To complete the first track you simply have to ensure that the gear you are currently in (out of three) is correct for the slope you are travelling on. And that the bike you have the pretence of controlling does not exceed one of the two extremes of speed – stalling, or going so fast you end up flying over the handlebars.

On the second dirt track you hit your first real obstacles, and find a use for the other controllable movements, the raising of the front and back wheels – not at the same time I hasten to add. This track also teaches you the premise upon which this game is based, precise control over the bike's speed. When travelling over stumps and rubble you must ensure that your speed does not exceed a narrow set of limits. Failure to do so results in the action freezing and a message informing you whether you went too fast or too slow. You may note that you do not even have the satisfaction of seeing your biker bite the dust.

By the time you have reached the third track you must have mastered the raising and lowering of the wheels – if not you might as well turn your computer off and start again. Here the main obstacle comes in the form of small holes, which require you to raise your bike wheels whilst passing over them. Next come three tracks with Beetles, the car variety, skips and watery pits. On these courses the control over your speed becomes too critical for your joystick to handle with any degree of accuracy, and you inevitably crash.

If speed and gear selections were all you had to worry about, then life might even approach something like simple, but you also have to ensure that you stay on the course as indicated on a small strip of the screen which shows sideways movement.

If you are a masochist who enjoys the sound of revving engines then you might enjoy this game – but I doubt it.

Adrian Pumphrey

***Gremlins Graphics,** once part of the Birmingham Software Syndicate, are to return to Sunny Sheffield where, hopefully, new ideas will reign.*

INFO 64

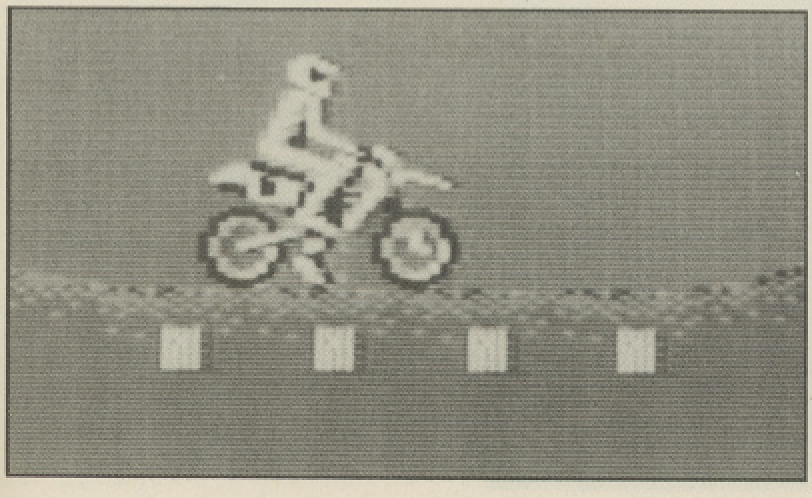
Gameplay: 39%
Graphics: 83%
Sonics: 81%
Lastability: 52%
Overall: 54%

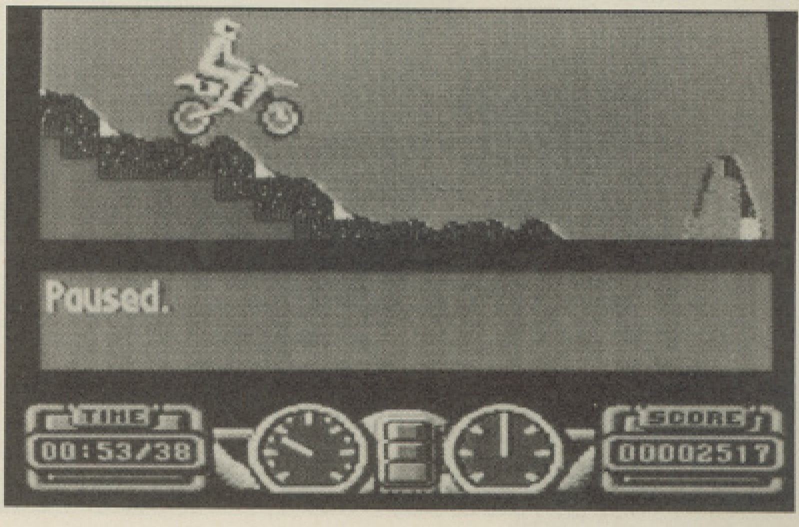
SUPER SCRAMBLE SIMULATOR

Gremlin – £9.99/£14.99

Mount up and pit your sanity against the latest game to bring you all the thrills and spills of high risk sport without the risks and without the thrills and even less of the sport.







RED STORM RISING

Microprose - PC - £34.99

Tom Clancy's novel of World War III submarine warfare comes to life as you take the helm of a US Attack submarine. Sooner or later, it had to happen.

WHEN Islamic fundamentalists destroyed the Russian oilfields, war looked inevitable as the Great Bear set its sights on the Middle East. The combat that followed as the super powers clashed would be partially decided by naval actions in the crucial North Atlantic. As a submarine commander your strategy and skill will be vital in hunting and destroying enemy subs, task forces and troop carriers.

Before you take on such responsibilities you can test your skills in a series of training missions, during which time you can learn how to use the sea's thermal layer to avoid enemy detection, the importance of keeping your speed low and a reasonable depth to ensure your propeller doesn't cavate in the water (and attract enemy sonar), and how and when to use the weapon systems. The toughest part of submarine warfare is tracking down your enemy without him finding you. This means you can't use active sonar, which would give away your position. You must find a quiet part of the ocean and listen for the signs that your passive sonar can find. Once you have a good sonar contact (above 90% - many torpedoes are fired in haste at poor contacts) you can fire your torpedoes at a submarine, or missiles at the surface contact.

Although you can play individual scenarios, the real challenge is when you play the Red Storm Rising Scenario and play your part in World War III. Here your missions are carried out and directly affect the backdrop of the war. Graphic sequences set the scene as Nato battles against the Soviet onslaught. An example of this is when you're sent in to intercept a task force carrying troops. Not only do you have to track down and attack the convoy, your success at sinking the troop carriers will decide the success or failure of the Soviet attack. If you do badly, the map will rapidly turn red.

As an idea, Red Storm Rising can't be faulted. It's based on a best selling book by a bestselling author and the computer implementation provides all the options and problems that face a modern submarine commander, who must do his best to win battles that effect the outcome of the war. Unfortunately, the game falls down on presentation. Although the clips showing the progress of the war are good, the actual screens used in the battles (most of the game) are dull. In fact, there's no comparison to games such as Silent Service or 688 Attack sub. As a result this will deter many gamers from a challenging and fascinating game.

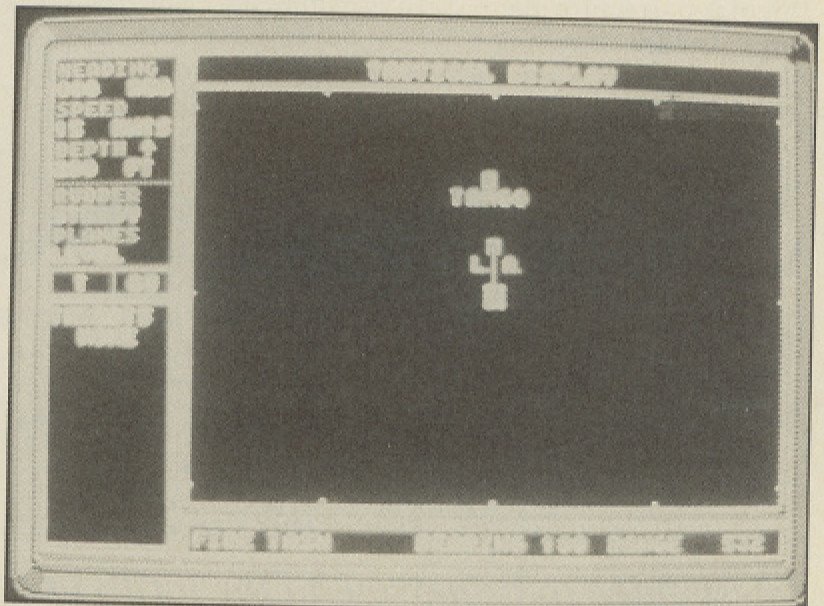
Tony Hetherington

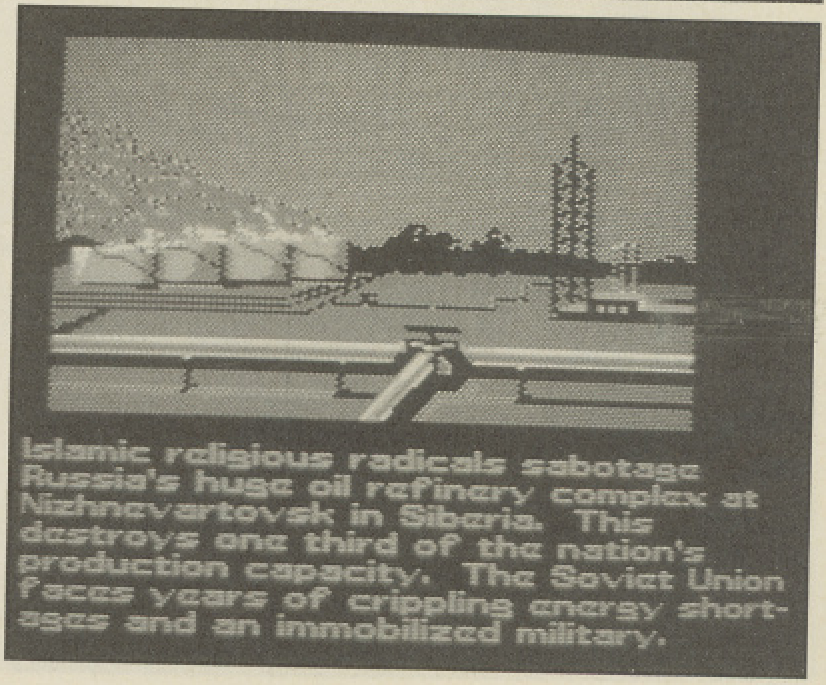
Red Storm Rising was written, in conjunction with Tom Clancy, by ***Sid Meir*** *who was the author of F-15 Strike Eagle and Silent Service. He was also the co-founder of Microprose.*

INFO

PC

Gameplay: 84%
Graphics: 32%
Sonics: 28%
Lastability: 67%
Overall: 53%





STORMLORD

Hewson - £14.99 Disc, £9.99 tape.

Fairy folk and foul creatures clash in a game that aims to take your screen by storm, as the heroic Stormlord rouses rebellion against a malevolent monarch.

AN evil Queen dominates the once beautiful and peaceful land. It was a land where fairies frolicked and butch heroes lay around being butch. It was also incredibly dull, so it's just as well that the Queen arrived with her evil minions. Hideous creatures roam the land, man-eating plants have taken root, and frolicking fairies lie captured in cages. Now Stormlord must rescue the fairies and liberate the land by destroying the Queen.

The land is represented on screen by a series of levels, each consisting of a number of sideways scrolling screens.

Docile plants, statues, steps, and boulders form the platforms on which our hero can walk and jump. Control is somewhat limited although you can build up power for a bigger leap by holding down the fire button. The screens also contain a very useful but inexplicable network of springboards that can hurtle our hero quite remarkable distances, and quite often land him by a damsel in distress. You can also be sure that there will be another springboard strategically placed for the return journey.

While Stormlord is running, jumping and leaping about the land various nasties are out to get him. Huge worms want to nibble him. Dragons want to fry him, flies want to swat him and Venus fly traps want to chomp him. However, he can hit back with a self loading, flying sword, but it's so ineffective that it's best to get out of the way.

There are also a number of traps such as boulders and acid which drop down from above and doors which block your path throughout the game.

Luckily, there are useful objects scattered around the landscape, but our brave and incredibly butch hero can only carry one at a time, which I suppose isn't that surprising when you consider that a key is almost as big as the door it opens! Apart from the keys, there are other objects whose function is less obvious, which can make it difficult to always carry the one you need next, resulting in quite a bit of backtracking. This can be reduced by leaving them near springboards.

The result of all this is a game that's a cross between a sideways scrolling shoot-em-up and a platform game. Although I doubt this will appeal to either game's supporters as it is rather silly and equally uninspiring.

Tony Hetherington

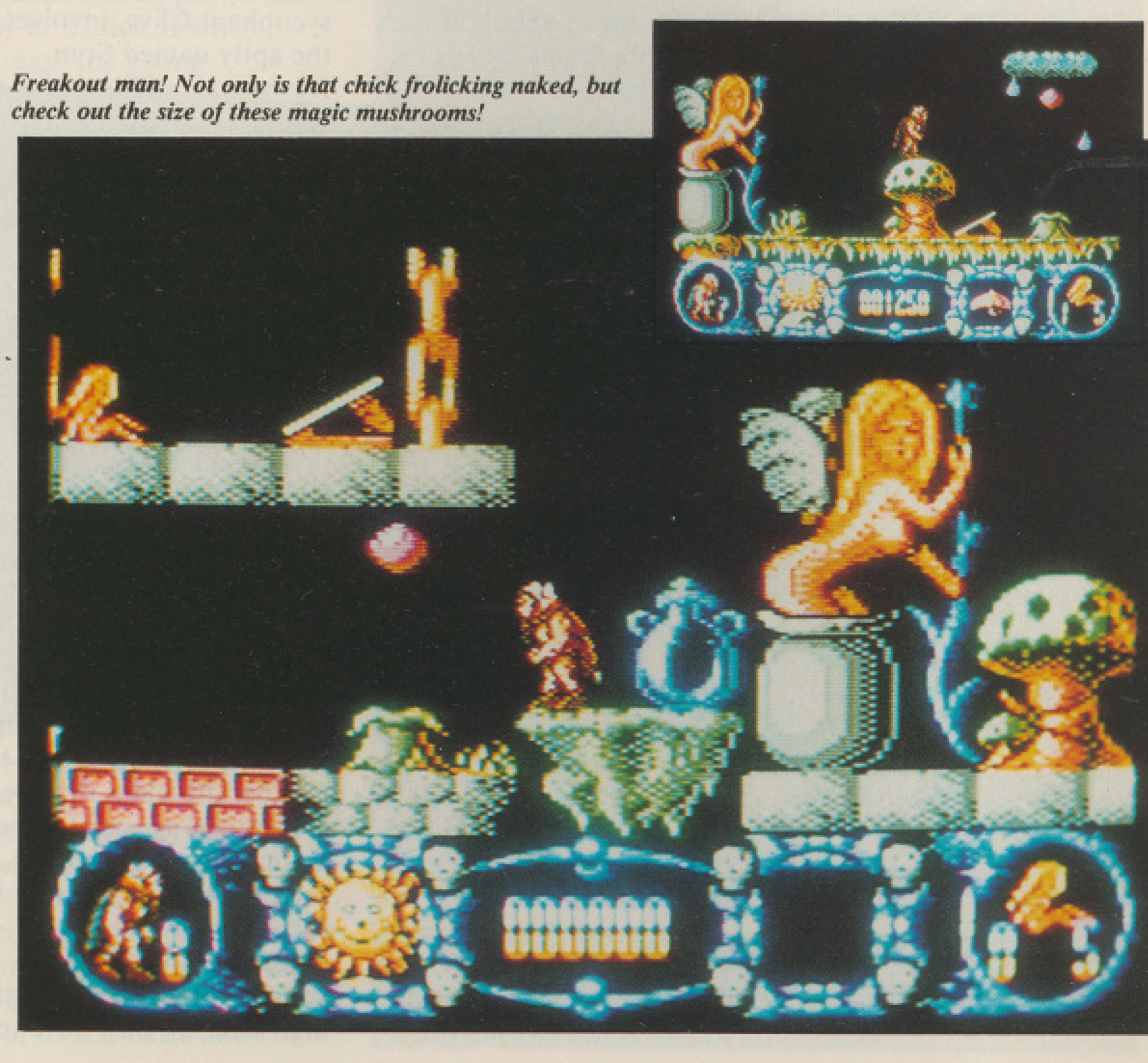
INFO

64

Gameplay: 37%
Graphics: 45%
Sonics: 52%
Lastability: 28%
Overall: 41%

*Stormlord was originally written by **Raffealle Cecco** (C64 conversion by Nick Jones), who also produced Cybernoid, Cybernoid II and Exolon.*

Freakout man! Not only is that chick frolicking naked, but check out the size of these magic mushrooms!



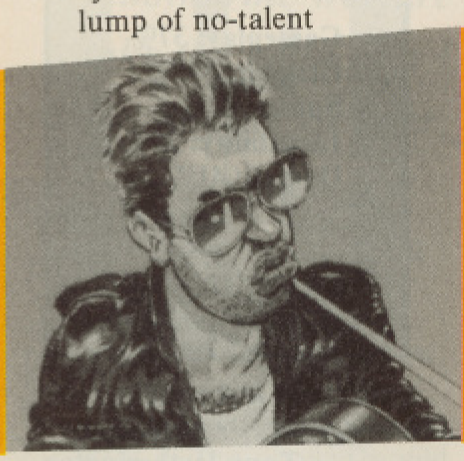
ROCKSTAR ATE MY HAMSTER

Codemasters – £9.95 Tape

Sex, drugs and Rock 'n' Roll – one of these is featured in Codemaster's latest full price game. Does the game live up to its rather peculiar name?

Gameplay: 90%
Graphics: 65%
Sonics: 80%
Lastability: 85%
Overall: 80%

WITH all the sweat and toil of making it to the top, Rockstar Ate My Hamster provides a gritty insight into the topsy turvy world of rock 'n' roll management. Through the eyes of a somewhat slimy lump of no-talent arrogance, called Cecil Pitt, you act as a manager, ducking and diving and calling the shots on the bands's future plans. But getting to the top is not so simple.



The ultimate goal of Rockstar, which has the intriguing feeling of a board game, is to go Platinum by topping the charts, the world and elsewhere. With your loutish sidekick Clive, who insists on using groovy talk, your first task is to choose the band. You are allowed up to four members and can select from a list of about 50 stars who appear one by one on a video screen in front of you. It is probably best to choose the maximum of four band members as somewhere along the road tantrums creep in and the odd terrorist attack tends to deplete the ranks. As for selecting stars, which are loaded from a data cassette, it varies from player to player depending on who can handle the big money stars and bigtime budgets.



Each star bears a distinct likeness to some of the big names in the pop world – including Bimbo Baggins (Kylie Mynogue), Dross (Bros) and many more. With your band together it's time to buy the gear. New equipment costs money but it's one step to true professionalism, with amplifiers that go to eleven for that extra push over the cliff. However, being one louder is not to everyone's taste so there is the option of second-hand gear, or even lower to the dodgy knock-offs.

This is where your climb to the top or low-life stroll into the wilderness begins. To start with, you have four options; practice, gig, publicity or gifts. It is probably advisable to lock them away for a bit of a jam, give them some publicity and then hit the road. The practice sessions cost money, so you have to choose your studio time wisely. The publicity, which is dubiously handled by the sycophant Clive, involves the aptly named Stun Newspaper and includes a subtle reworking of that infamous headline concerning a certain Starr eating a hamster. Publicity stunts, however, do not always get covered by the press, but that's a chance you have to take.

If you decide to gig you have to select a venue from the list of pubs, clubs, Universities, Concert Halls and stadiums. For the first time out it is probably best to play to cheap and easy-to-fill venues. A decision on the ticket price has to be reached and the number of nights playing. After each gig Clive informs you of the number of tickets that were sold and consequently the amount of money made. At the start of the game money is the key to success especially if you have chosen band members with high salaries, they have to be paid on a weekly basis.

If gigs are going well and you have succeeded in front page gutter news, the phone will start to ring with offers of record contracts, charity gigs and potential sponsorships. When a satisfactory record deal is reached you have the option of cutting the vinyl and shooting a promo to begin your claim on the record sales market and a position in the national charts. The prospect of shooting a video means more decisions on producers, locations and image details with every decision being of paramount importance for the band's future. With some suspiciously recognisable tunes the band under your guidance will enter the realm of the weekly charts and face the problem of staying at the top, paying taxes and sueing bootleggers.

Rockstar Ate My Hamster is an enjoyable game even though the graphics are not particularly good. It is addictive with reasonably good sonics for its calibre and price, and for this reason alone it is worth considering.

Mark Jones

Colin Jones, the programmer of all the formats of Rockstar, is undoubtedly, and without disgrace in any way, Welsh.

ULTIMA TRILOGY

Origin Systems (Microprose) – PC £29.95. C64 £24.95 disk only.

More than one million global gamers have delved into the Ultima series of role playing games. Now the first three are available in a special edition.

THE Ultima Trilogy is a special edition of the infamous Ultima games in which you must confront a Triad of Evil who threatens the land ruled by Lord British. Both Ultima I and III have been available in the UK before but this is the debut for Ultima II, which will provide many more hours of role playing whilst filling some of the gaps and answering some questions that were left open with its omission.

The content of the box is impressive and immediately gives the impression that this is a game to play. Alongside the game disks (3 for C64, 4 for PC) there is a quickstart guide, a player reference guide that outlines the keyboard controls and available spells, and a massive 100 page instruction manual that covers everything from the book of the amber runes to a guide to the monsters and critters you will face. It also contains three full colour maps of the lands you must save.

In the games you play a brave adventurer who may be of human, elf, dwarf or bobbit stock, and be trained in the art of a fighter, cleric, thief or wizard. Throughout the games your hero is controlled through a series of single key presses that correspond to commands. For example, 0 lets you offer gold as a bribe. U unlocks doors if you have the key, and I ignites a torch. Although you begin alone with a number of points to distribute between your role playing characteristics, you will soon find others that will join you in your quest.

Ultima I is the first age of darkness and features a desperate battle against hordes of nightmarish creatures spawned by the Wizard Mondain. The defenceless people of Sosaria flee in vain to the ancient strongholds and all will be lost if a hero is not found.

The same heroic abilities will be put to the test in Ultima II when Minax, the young apprentice of Mondain, seeks to avenge her father's death. Such is the power of this Enchantress that her efforts to find her father's slayer have torn rifts in the fabric of time. This has opened doors through which a brave adventurer might bring about her doom.

In Ultima III the third part of the Triad of evil is revealed, as fragments of a manuscript hint at an unholy alliance between Mondain and Minax and of a demonspawn that their union had produced. This is the ultimate test for a hero as evil openly stalks the land and your foe is unknown apart from the power it possesses.

The games look a little dated now compared to the more recent Ultima IV and V, and other games such as Pool of Radiance and Bard's Tale III, but this special edition is a must for Ultima players looking to complete the story.

Tony Hetherington

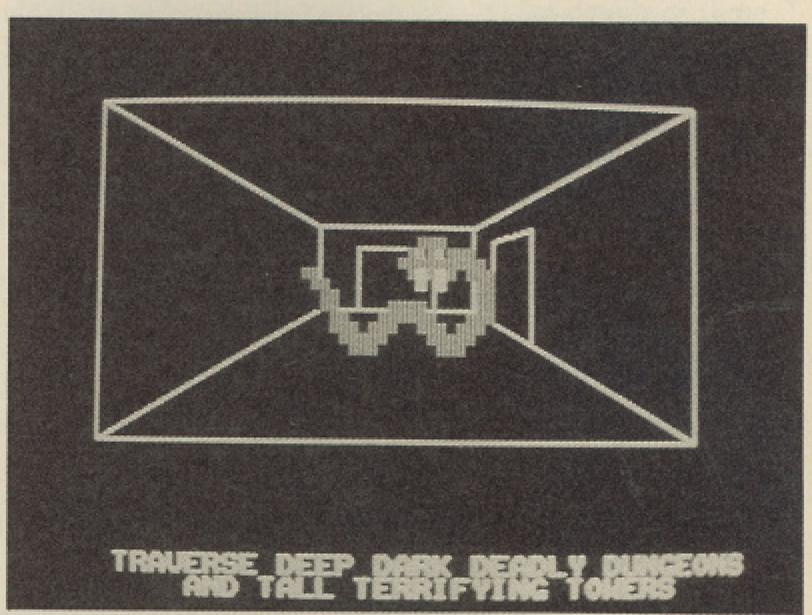
INFO – PC

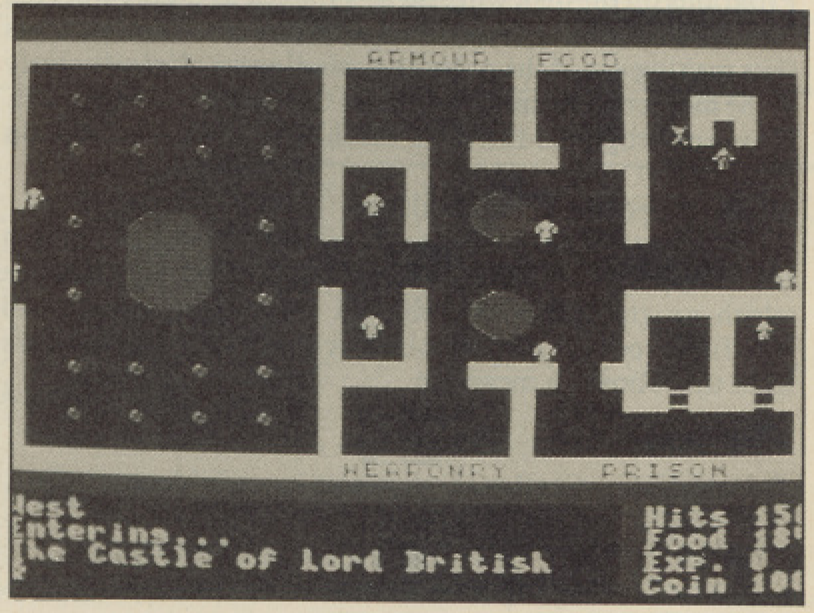
Gameplay: 88%
Graphics: 45%
Sonics: N/A
Lastability: 85%
Overall: 72%

INFO – C64

Gameplay: 78%
Graphics: 55%
Sonics: N/A
Lastability: 80%
Overall: 71%

*The Ultima adventures were written by **Richard Garriott** between 1979 and 1988 firstly being published through Sierra-on-Line and then through Origins, the company he co-founded. The first Ultima game to appear in the UK was imported by US Gold (Ultima III) since then others have been imported by Microprose.*





Software for Sale

If you think that one of our programs looks very interesting, but you can't afford the time to type it in, then our software service will help you out

It's three o'clock in the morning. You sit at the computer keyboard having just finished a marathon typing session entering one of the superb programs from *Your Commodore*. Your fingers reach for the keyboard and press the letters R, U and N. You press RETURN, sit back and nothing happens.

Everyone has probably faced this problem. When it does happen it's a matter of spending hours searching through the program for any typing mistakes. No matter how long you look or how many people help you, you can usually guarantee that at least one little but slips through unnoticed.

The *Your Commodore* Software Service makes available all of the programs from each issue on both cassette and disk at a price of £6.00 for disk and £4.00 for cassette. None of the documentation for the programs is supplied with the software since it is all available in the relevant magazine. Should you not have the magazine then back issues are available from the following address:

INFONET LTD, 5 River Park Estate, Berkahmsted, Herts, HP4 1HL.
Tel: (04427) 76661

Please contact this address for prices and availability.

The Disk

Programs on the disk will also be supplied as totally working versions, i.e. when possible we will not use Basic Loaders thus making use of the programs much easier. Unfortunately at the moment we cannot duplicate C16 and Plus/4 cassettes. However programs for these machines will be available on the disk.

What programs are available?

At the top of each article you will find a strap containing the article type, C64 Program etc. So that you can see which programs are available on which format, you will also find a couple of symbols after this strap. The symbols have the following meaning:

This symbol means that the program is available on cassette.

These programs are available on disk.

Please Note

Since the programs supplied on cassette are total working versions of the program, we do not put disk-only programs on tape. There is no sense in placing a program that expects to be reading from disk on to tape.

APRIL 1989

BASEX – Give your C64 new sound, graphics and toolkit commands as well as a machine code assembler with this Basic extension.
AUTOSCROLL – Professional text screen scrolling with this C64 utility.
BALANCE SHEET – Keep your bank manager happy by keeping better track of your money with this C64 program (disk only).
ICON 64 – Add icon control to your own programs with this C64 utility.
80 COLUMNS – A suite of Hi-res graphic commands for C128 80 column users.
ZEUS – A disk file copier for Plus/4 users. Available on disk only.
MULTI-PRECISION ARITHMETIC – A suite of powerful mathematical routines expanding your C64's number crunching abilities.
STORAGE SPACE – Use the RAM behind the C64's Kernal and Basic ROMs for data storage (available on disk only).

ORDER CODE
DISK YDAPR89 £6.00
TAPE YCAPR89 £4.00

MAY 1989

ANTI-FREEZE – Protect your C64 programs from cartridge-based freezers.
MULTI-COLOURED LIST – Brighten up your C64 Basic listings by adding on-screen colour.
PRINTFX – A suite of powerful screen printing commands to add to your Plus/4's Basic. (disk only).

BANKER 128 – Money Management for C128 disk users.
TURBO TOUCH – A superb typing trainer for C64 disk users.

ORDER CODE
DISK YDMAY89 £6.00
TAPE YCMAY89 £4.00

JUNE 1989

RETURNER – A handy routine that allows your C64 programs to re-load a disk based menu. (disk only)
LINE INPUT – Improve your C64's INPUT routine
PROGRAM ANALYSER – A suite of programs to help you examine your C64 Basic programs in depth.
ASM ASSEMBLER – A powerful 6510 assembler for the C64

ORDER CODE
DISK YDJUN89 £6.00
TAPE YCJUN89 £4.00

JULY 1989

MOUSE 80 – Use a 1351 mouse in 80 column mode on the C128.
QUICK SEARCH – Add a FIND option to C64 Basic.
SPRITE LIBRARY – Add a collection of circles to your growing sprite library (C64)
HEAD FOR HOME – An update on our popular C16 and Plus/4 game. Available on disk only.
WINDOWS ON THE 64 – Powerful windowing commands for the C64.
SPRITZ SPRITE – Give your Plus/4 or expanded C16 sprites – just like a C64. Available on disk only.

ORDER CODE
DISK YDJUL89 £6.00
TAPE YCJUL89 £4.00

AUGUST 1989

REASONING ON THE 128 – Programs for the first part of our expert system series (128 Disk only).
THESAURUS – Build up your own C64 thesaurus with this program for disk or tape.
WINDOW ON A MAZE – Generating mazes on your C64
ADDRESS BOOK – A cassette based address book for C64 users
DISK SCRAMBLER – Protect your disk data from prying eyes (C64 disk).

ORDER CODE
DISK YDAUG89 £6.00
TAPE YCAUG89 £4.00

Cassettes or disks are available from March 1986. Please ring the editorial office (01-437 0626) for details of these.

ORDER FORM – PLEASE COMPLETE IN BLOCK CAPITALS

NAME	QTY	TAPE/DISK	ORDER CODE	PRICE
SEPT 1989		TAPE (£4.00)	YCSEP89	
SEPT 1989		DISK (£6.00)	YDSEP89	
OVERSEAS POST £1				
			TOTAL	

NAME...

ADDRESS..

POSTCODE...

I enclose a cheque/postal order for £....... made payable to ARGUS SPECIALIST PUBLICATIONS LTD.
All orders should be sent to: *YOUR COMMODORE,*
READERS SERVICES, ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD, HERTS. HP2 7ST

Please allow 28 days for delivery.

TERRAMEX

Grand Slam – £2.99 Tape

High in the icy vastness of space, something is headed our way. A giant asteroid is on collision course with Earth and only you can save the world.

WELL actually the only person who can save the world is Professor Eyestrain, who predicted the asteroid's coming some 20 years ago. The trouble was that no-one believed him, so he went off to a dark and dingy continent to sulk.

Taking the part of one of five stereotyped explorers you must search out the lager of Lamot, and then search out Professor Eyestrain so he can build an asteroid deflector. Although if the world is saved then Grand Slam will be able to write more programs like this.... Hmm, total destruction might not be so bad an idea after all.

When Terramex first appeared on the ST and Amiga it had the feel of an average 8-bit game, and now we see it in budget 8-bit form, which is better value for money. It doesn't make the game any more interesting though.

Having selected your character and waited for the rest of the game to load, you set off into the dark continent followed by your invisible bearers, in undramatic 2-D arcade adventure style. An irritating piece of music will have your teeth grating by now, so go for the volume switch whilst concentrating on the smallish but colourful characters, and the generally dull landscapes.

One of the first objects encountered, and this game is object intensive, is a vacuum cleaner, which endows the power of flight. Yes, you'll believe a hoover can fly, but not everywhere. One of the failings of Terramex then becomes obvious, you can hoover along a certain distance and suddenly because it would make the game too easy, you stop in mid air, unable to progress further. There isn't actually anything there stopping you from going on, you just can't. One of the next objects needed is an umbrella which acts as a parachute. One good point is that you can ask your character for an idea when faced with a problem. If you're lucky the right object will flash up, and then it's down to you to use it properly.

Besides the object manipulation part, there's also the challenge of having to leap over snakes, crevices, and other obstructions, in an Indy Jones like manner. Should all efforts fail, then a quite humorous sequence follows whereby the asteroid ploughs into the Earth. For sheer awfulness this takes some beating.

For aficionados of the arcade adventure, £2.99 represents good value, if unspectacular entertainment, but for those not converted to the cause, there is nothing here to convince that an expedition to find Professor Eyestrain is a better idea than waiting at a bar for disaster to fall.

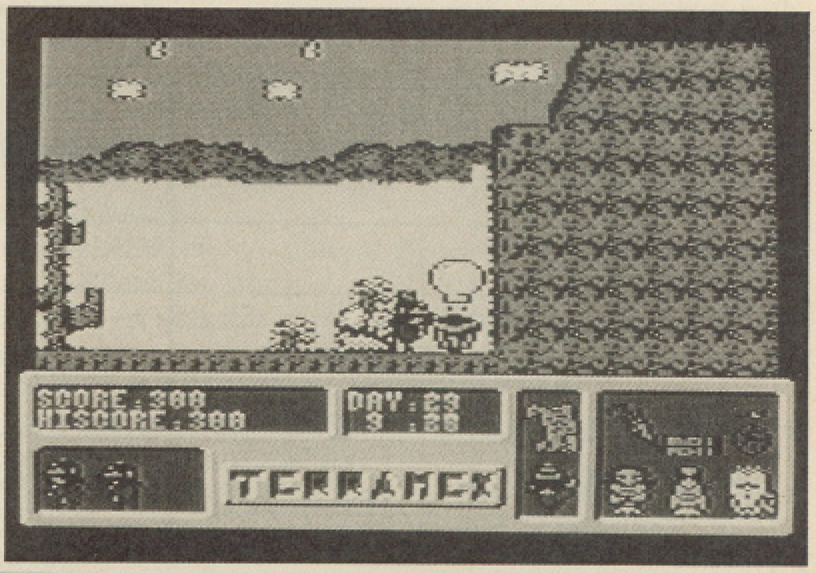
Duncan Evans

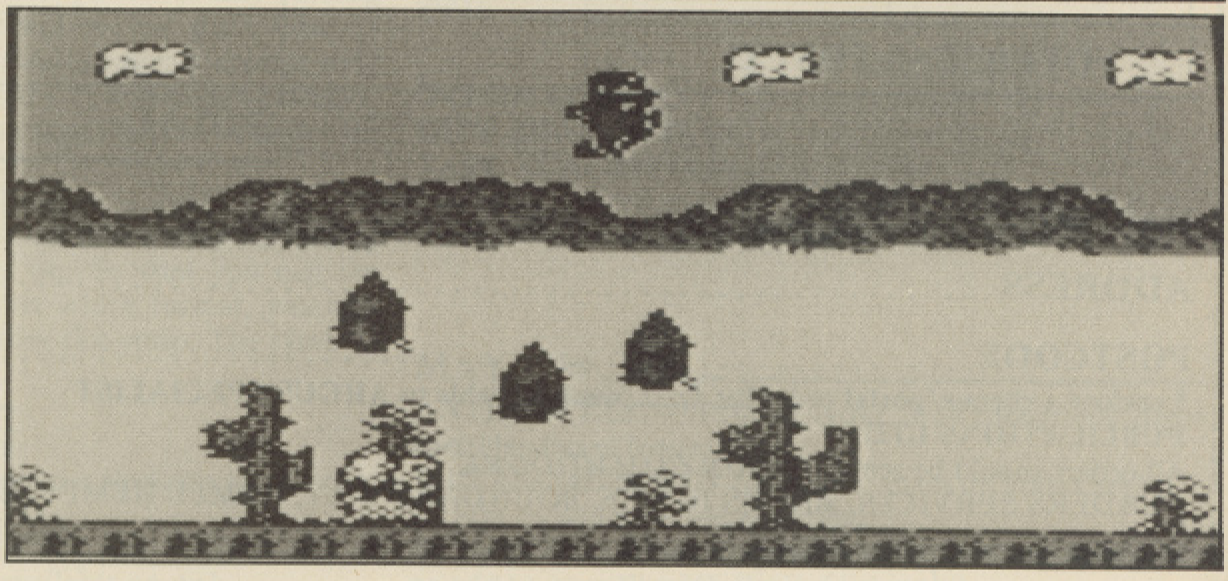
INFO

64

Gameplay: 68%
Graphics: 72%
Sonics: 67%
Lastability: 71%
Overall: 69%

Grand Slam *started life quite shakily, with Terramex being one of its poorer releases. Recent offerings are generally much better.*





FIRE KING

Electronic Arts, Price £18.99 Disc

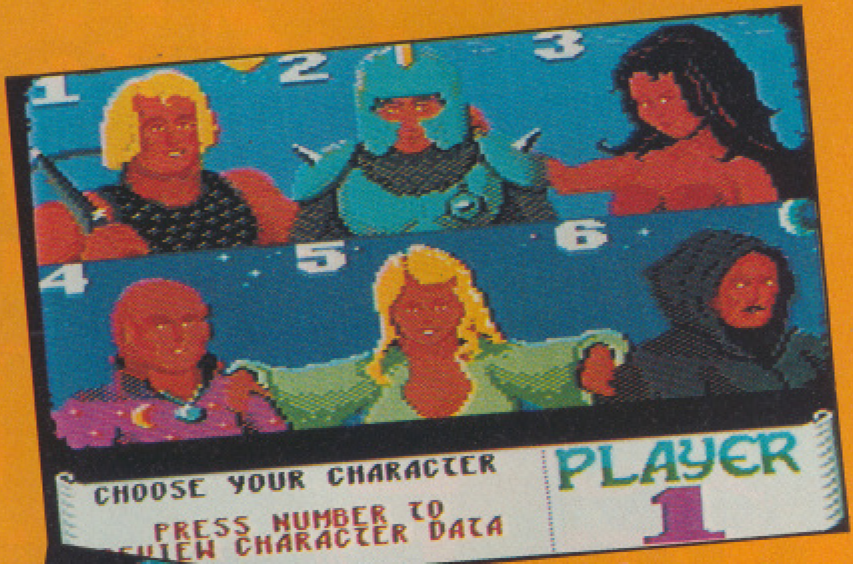
Role-playing down under is the name of the game. Gordon Hamlett casts a clairvoyance spell and determines if everything is fair dinkum or not.

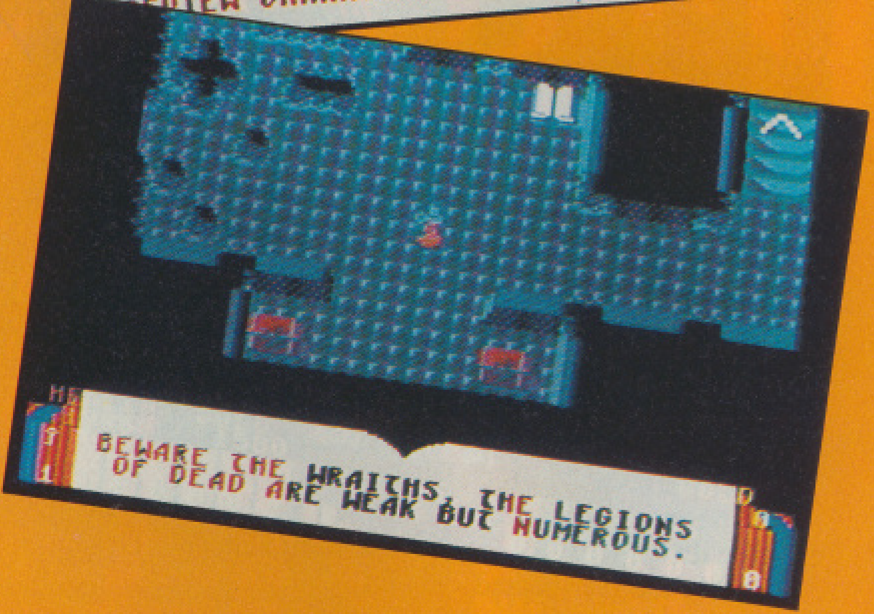
THE land has always been ruled by one of the four elemental magi. At the time of the great conclave, the Fire Wizard happened to be king, but news soon spread through the town that he has been horribly murdered. The Earth Mage has fled, the Water Mage has disappeared and the Wind Mage has been grievously wounded. Rumours of an enormous magical beast abound as food supplies start to run dangerously low. In desperation, the mayor sends forth a hero who promptly disappears. Now, you have been summoned to help.

You have the choice to play one of six characters, each differing in their strength, armour and magical abilities. These attributes can be altered temporarily with the help of magic or permanently by finding relics – something that you will have to do if you are to get very far in the game.

Fire King follows the usual role playing ideas in so much as you have to wander round fighting monsters and finding key objects. There are books scattered around containing clues, although some information, especially in the Thieves Guild, has to be paid for.

The key to the game lies in manipulating the objects that you find and/or purchase. You have seven pockets each of which can contain up to nine similar objects, eg. keys, death spells, etc. The problem is that there are usually more than seven types of objects to carry, so you will need to plan ahead. On top of that, there is food to buy which will help to top up any energy lost in battle.





Graphically, the game looks and plays a lot like Gauntlet with the top down view and monster producing vortexes. On top of that, there is more of a feeling of adventure to the game rather than just one large shoot-em-up.

The problem with Fire King is the manipulation of objects. The pocket system is very cumbersome to use, especially in a 'real time' situation. The monsters are everywhere and I suspect that fans of role playing games will find this too much of an arcade game for their liking. I must confess that knowing their previous track record, I expected something a bit more cerebral from this company.

Gordon Hamlett



INFO

64

Gameplay: 48%
Graphics: 51%
Sonics: 41%
Lastability: 58%
Overall: 50%

*Written by an Australian Company called **Micro Forte** and Licensed through SSG – their first venture away from wargames.*

PHOBIA

Mirrorsoft – £14.95 Disc, £9.95 Tape

I suffer from Scrollaphobia, fear of horizontally scrolling shoot 'em ups, so when the latest game from Mirrorsoft arrived I broke into a nervous sweat.

NOT another scrolly shooter I thought as the Deputy Editor handed me the sealed package with a smirk. I could hear the sounds of shooting, see the disc writhing about in the Jiffy before I even opened it. I gritted my teeth as I removed the staples (and sent them back to Rik – he's very mean you know), and the disc leaped out, wobbling manically, into the disc drive.

Slowly the fear passed and I slipped into my reviewer alter ego, sloping forehead, protruding teeth, beer belly flobbing over green cords, joystick warming in sweaty hands. Scroll, shoot, kill, nyarrgghh!

But wait, hold onto your jockstraps, Phobia is different. Slightly more than not much different, but different enough.

There's strategy. The evil Phobos has captured the Galactic Emperor and you've got to rescue him. Why? Well don't ask silly questions. You need to navigate nine of the fifteen Phobia planets to obtain a piece of heat shield, with which you can attempt to rescue the Emperor from the sun where he's being held. Along the way, the minions of Phobos will be seeking to stop you, constructing barriers which can only be removed by sacrificial pods obtained from the moon. The trick is to stay one step ahead of the bad guys and avoid having to collect pods every five minutes.

And there's the psychology. All the Phobia planets feature a specific distorted form of reality based on human fears. Fear of water, spiders, enclosed spaces (you can guess how easy this one is), dentists, and even death. I didn't see Payolaphobia, fear of not being paid, so I hope the YC chaps don't slip that one in (it slipped – Dep. Ed).

Then there's the action. Heaving colours and scenery, scrolling from right to left, very smoothly considering just how colourful and action packed the screens are.

There are power ups along the way, including pods, and a simultaneous two player mode, which can be slightly confusing at times, that guarantees an intense shooting experience. The graphics are very good, and are complemented by the excellent music and decent crunchy bits.

Phobia isn't so difficult you never want to play again, but isn't easy enough to play while eating a spaghetti sandwich. I was particularly impressed by the Death planet, with skeletons in coffins, and corpses swinging from ropes attached to the ceiling, and even the end of the level confrontations have been designed to send a shiver up your spine.

Just when you thought it was dull to go for a scroll, Phobia puts the thrill back in the kill, and laid this reviewer's Scrollaphobia to rest, by delivering a nose crunchingly exciting thrash.

Duncan 'barking mad' Evans

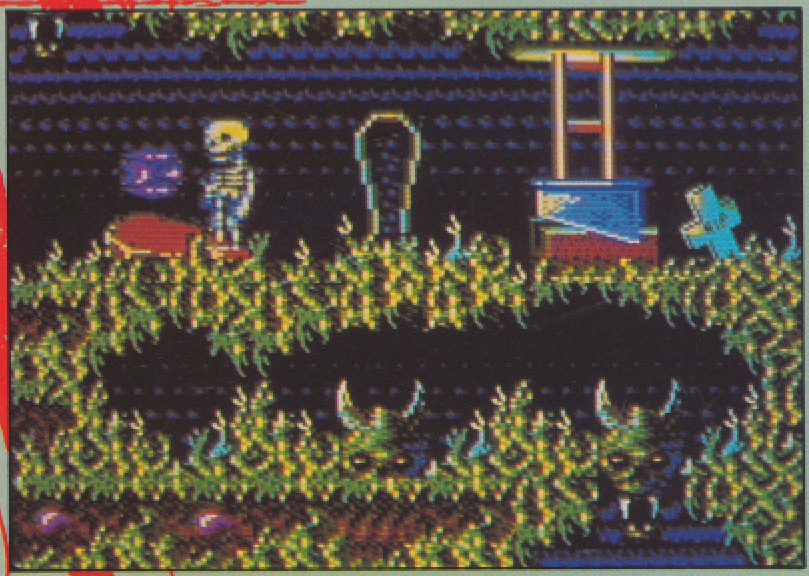
***Tony Crowther** is a legend in games writing, which means he hasn't done anything good for ages, but this is the big one.*

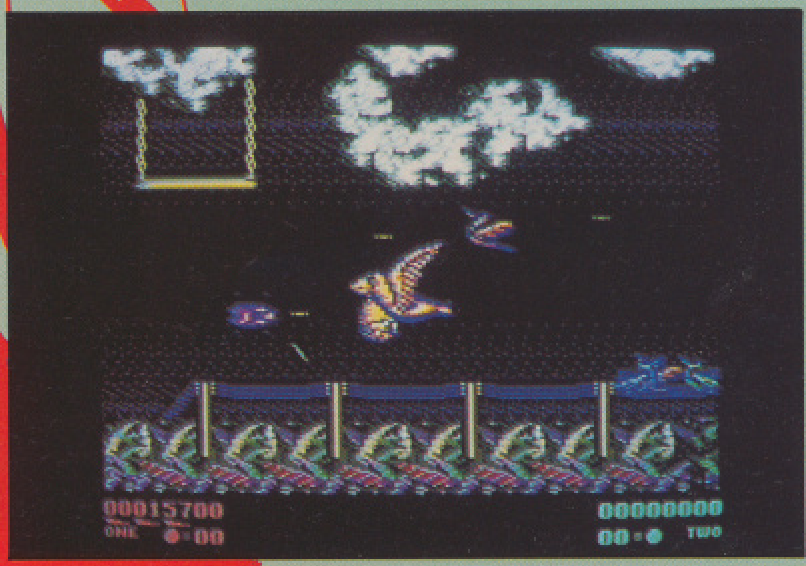
YC SURE SHOT

INFO

64

Gameplay: 90%
Graphics: 96%
Sonics: 88%
Lastability: 86%
Overall: 92%





JOURNEY TO THE CENTRE OF THE EARTH

US Gold – £19.99

Face danger, excitement and the unknown as you attempt to follow in the steps of 16th Century explorer Anre Saknussen, that lead to the centre of the Earth.

IF you were to imagine a game based on Jules Verne's Journey to the Centre of the Earth, programmed by the team that wrote Joan of Arc, you certainly wouldn't imagine this game. Despite the fact that the game encompasses the many features ideally suited for a role playing, strategy or adventure gaming, the actual mix of restricted strategy and randomly generated events is incredibly disappointing.

The game begins quite well with an impressive demo of the sort of music and graphics even a CGA PC is capable of, but if you pardon the pun, it's all downhill from here. You begin the game with a choice of four explorers and an expert guide, although after a rockfall you're on your own. The rockfall is the first of the "events", but more of this later.

Rapidly, the game slots into its humdrum routine based upon a screen showing the full and incredibly limited map that you have to navigate, displays showing your food, water, vitality and physical condition. There are also a series of icons, which must be selected to move, rest (to conserve energy), ration (to spread your food further) and one that leads to the medic screen. Here you can get a diagnosis of your ills, and there's a first aid kit to patch them up.

Whenever you click the move icon you hear a few quiet footsteps and something may happen, ranging from a silly event to an even sillier arcade sequence. For example events include snakes that dart in, bite you and depart; pebbles that trip you; and bumps on the head that turn into fractured skulls just because of a badly placed stalactite. As for the arcade sequences, try these two for examples. In the repetitive falling rock game you have to run up three slopes while avoiding the rocks. Yes, you read this correctly, you run up even though you're going down into the Earth! The second example is when you have to avoid a herd of stampeding elephants! Exactly what a herd of elephants is doing beneath the earth's surface isn't explained, neither is how the thin tunnel you were in suddenly becomes large enough for a conglomeration of mammoths.

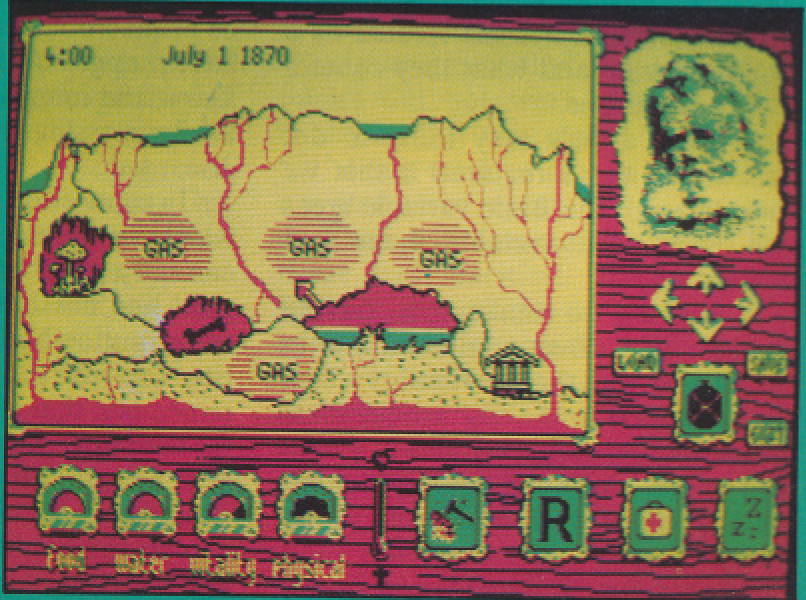
I find this the most annoying type of game as it promises so much, looks and sounds so good and delivers so little.

Tony Hetherington

INFO

Gameplay: 19%
Graphics: 67%
Sonics: 63%
Lastability: 20%
Overall:
42%

*Journey to the Centre of the Earth is produced by French developers **Chip**, who also wrote Joan of Arc marketed by US Gold.*



Extending Basic

Get a printout of the variables of your Basic program with this latest routine in our series on extended Basic.

By Burghard-Henry Lehmann

In this series of articles on developing extended Basic routines, we have already dealt, several times, with the way Basic stores its variables. Therefore, the next logical step is to write a routine which gives us a printout of the variables created by a Basic program and what they contain at any given time.

This is a very easy thing to do, because all that's required is that we loop through the Basic variable areas and print each variable name and content.

I wanted to make this facility possible as a direct command and in connection with the trace facility which we developed in the last article. I am including this in this instalment for all you people who didn't buy last month's YOUR COMMODORE (shame on you!).

As a direct command, you simply enter VARS, and all variables of the Basic program which you have just run are printed out. Of course, if you haven't run a Basic program or have given the direct CLR command, which clears all Basic variables, nothing will be printed out. The same is true if you have changed the Basic textfile. As you know, the moment you enter a Basic line by pressing return all variables are discarded.

In connection with the trace facility which we developed last month, you can call a printout of all variables after each line has been executed and the trace routine has reprinted the line at the top of the screen. You will remember, that after each tracing step, the computer waits until you press any key. To get a printout of the variables simply press 'V'. If you want program execution to continue press any other key.

How Basic Stores Its Variables

Before we look closer at the program itself, let's briefly recapture how Basic stores its variables.

All Basic variables are stored directly after the end of the Basic textfile. To find the start of the variable area you have to peek the zero page variable decimal 45/46, Hex 2D/2E. Each variable takes up seven bytes of memory.

The first two bytes contain the variable name. The way the variable name is entered shows what type of variable we are dealing with:

A floating point variable has both bytes of its name entered in simple ASCII. If the name consists of only one letter, the second byte contains zero.

An integer variable has 128 added

to both its name characters, and if there is only one character (which, as you remember, always has to be a letter!), the second byte contains 128. (All this means, that in each case, bit seven of the eight binary bits is set.)

A string variable is distinguished by having the letter of its name entered in simple ASCII, while the second character has 128 added to the ASCII value, or 128, if there isn't a second character.

Finally, a function definition has 128 added to the first letter of its name, while the second character is simple ASCII or zero. (Incidentally, in the program given I have ignored function definitions. As always, this is partly due to my laziness and partly in order to encourage you to add this to the routine yourself...)

The final five bytes of each variable contain the contents of the variable itself. This is easiest in the case of the string variable: Here, the third byte contains the length of the string and the fourth and the fifth byte contain the base address of the string in the Basic textfile.

This is quite a clever way of going about things. Instead of storing the string itself in the variable area, which would of course take up oodles of memory and make the whole variable area pretty difficult to manage, a pointer address which takes up only two bytes is stored in the variable area. The disadvantage with this method is of course, that the Basic textfile must not be changed. The moment it is changed, the pointer to a certain string in the variable area might not be correct any more. This is one of the reasons why you can't continue running a Basic program after you have changed the textfile.

The storage of an integer variable is also pretty straightforward. The value of the variable is stored in the third and fourth byte after the name, whereby the third byte contains the high byte of the value, plus the so called sign bit, and fourth byte contains the low byte. You assign an integer variable by entering '%' after it, e.g. X%.

An integer ranges from -32768 to +32767. If you add those two numbers up you get 65535, which is 65536 short of one, because the zero is included as a valid number. And 65536 is the maximum range of a 16-bit address. If bit seven, or the sign bit, of the high byte is set, the number is considered to be negative, if it is clear, the number is considered to be positive. For example, high 0 and low 100 represent +100, while high 255 and low 156 represent -100.

The most complicated variables of all are the floating point variables. This is a pretty long subject which I do not want to enter at this point. There is also no need to, because in our program we are using ROM-routines to convert the floating point number as it is stored in the variable area into an ASCII string which we then simply print out.

But, for the mathematician amongst you, here is how a floating point number is stored in the Basic variable area: The byte after the variable name contains the exponent, the byte after that contains the first mantissa, plus the sign bit, and the final three bytes contain mantissa two to four.

Printing the Variables

The routine which prints each variable name and the contents of the variable, is so straightforward, that it needs very little explanation.

The main loop, which I call VARSLOOP, first of all prints the variable name and decides at the same time what type of variable has to be dealt with (lines 4180-4310). According to those tests the program flow then jumps to the appropriate routine: FLPOINT, if it is a floating point variable, INTEGER, if it is an integer variable, and STRING, if it is a string variable.

Before the value of each type of variable is printed, some spaces and an equal sign are printed, as well as the '$'-sign, in case of a string, and the '%'-sign, in case of an integer. This makes the printout more presentable.

In the case of a floating point variable, A is pointed at the low byte of the first byte of the floating point value and Y is pointed at the high byte (lines 4470-4540).

Then the ROM-routine at $BBA2 is called to put the value into the first of the two floating point accumulators.

Next, the ROM-routine at $BDDD converts the contents of the floating point accumulator into a printable ASCII string. On exit from that routine, A and Y point to the base of that string.

Finally, the string is printed in the usual fashion, with $AB1E, which prints any string point at by A/Y and terminated with a zero.

An integer variable is printed by loading its high byte into A and its low byte into X. The ROM-routine at $BDCD, which prints any number contained in A/X, prints the integer value out.

To print a string variable, we first get the length of the string and store it in 255 (line 5450-5470). Then we transfer the address which points at the string in the textfile into zero page 253/254 (lines 5510-5560).

Finally, we use indirect-Y to print the string, letter by letter. This is done with a loop which is limited by the length value contained in 255 (lines 5600-5650).

After a variable has been dealt with, VARSLOOP ends by incrementing the pointer address by seven (lines 4770-4830). The vital loop exit test, which tells when there are no more variables, is done at the beginning of VARSLOOP (lines 4070-4120). This is because the routine has to exit at once if there are no variables at all!

The end of the Basic variables area is contained in zero page $2F/30. If this point is reached, there are no more variables. What follows are the Basic arrays with which we are not dealing at this point.

Final Notes

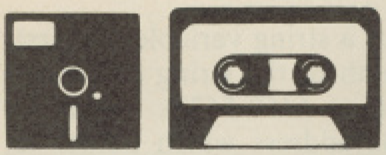
As I was developing the variables printout routine I discovered to my great puzzlement that certain variable names did not work properly. For example 'CC' was stored as 'C' and 'C1' didn't work at all and resulted in a syntax error report.

Then I discovered the reason for those irregularities: The extended Basic commands we use starts with a letter, that is a normal ASCII character. If you declare a variable by omitting the LET command (as one usually does), then the variable name too starts with a letter!

To overcome this I changed the main routine in the following way: First, I added two lines, which test for

a stolen character (lines 550-560). If a token has been found, which means that the line starts with a Basic command, the routine jumps straight to $A7ED which evaluates the Basic command in the usual manner.

Next, before starting the tests for our extended Basic commands, I save the Basic pointer $7A/7B (lines 600-630). If the tests are negative, that is, if none of our extended Basic commands has been entered, $7A/7B is recovered (lines 1070-1100). This resets the pointer to the first character on the line, resulting in a variable name being read correctly.



```
10              ORG 49152
20              ENT
30      ;
40      CHARGET    EQU $0073
50      EXECVECT   EQU $0308
60      PRINT      EQU $E716
70      PRINTNO    EQU $BDCD
80      PRINTSTR   EQU $AB1E
90      PRINTCR    EQU $AAD7
100     PLOT       EQU $FFF0
110     ;
120     ;
130     ;
140     ;TURN EXTENDED BASIC ON
150     ;BY CHANGING VECTOR AT $0308
160     ;
170     EXTBASON   LDA #<PRGSTART
180                STA <EXECVECT
190                LDA #>PRGSTART
200                STA >EXECVECT
210     ;
220                RTS
230     ;
240     ;
250     ;
260     ;TURN EXTENDED BASIC OFF
270     ;BY CHANGING VECTOR AT $0308
280     ;BACK TO NORMAL ($A7E4)
290     ;
300     EXTBASOFF  LDA #<$A7E4
310                STA <EXECVECT
320                LDA #>$A7E4
330                STA >EXECVECT
340     ;
350                RTS
360     ;
370     ;
380     ;
390     ;*** MAIN PROGRAM ENTRY ***
400     ;
410     ;LOOK FOR EXTENDED BASIC COMMANDS
420     ;
430     PRGSTART   JSR CHARGET
440                JSR EXECSTM
450                JMP $A7AE
460     ;
470     ;IF NOT DIRECT, DO TRACE.
480     ;
490     EXECSTM    PHA
500                JSR TRACE
510                PLA
520     ;
530     ;IF BASIC TOKEN, GO NORMAL.
540     ;
550                CMP #128
560                BCS NORMAL1
570     ;
580     ;SAVE $7A/7B.
590     ;
600                LDY <$7A
610                STY <251
620                LDY >$7A
630                STY >251
640     ;
650                CMP 'O
660                BNE NEXT
670                JMP OFF.RT
680     ;
690     NEXT       CMP 'C
700                BNE NEXT1
710                JSR CHARGET
720                CMP 'O
730                BNE NEXT1
740                JSR CHARGET
750                CMP 'L
760                BNE NEXT1
770                JSR CHARGET
780                CMP #$B0     ;'OR' TOKEN
790                BEQ COLOR.RT
800     ;
810     NEXT1      CMP 'V
820                BNE NORMAL
830                JSR CHARGET
840                CMP 'A
850                BNE NORMAL
860                JSR CHARGET
870                CMP 'R
880                BNE NORMAL
890                JSR CHARGET
900                CMP 'S
910                BNE NORMAL
920     ;
930     ;CALL PRINT VARIABLES ROUTINE.
940     ;
950                JSR VARS.RT
960     ;
970     ;GET NEXT CHARACTER.
980     ;
990                JSR CHARGET
1000    ;
1010               JMP NORMAL1
1020    ;
1030    ;
1040    ;
1050    ;DO NORMAL ROM-ROUTINE.
1060    ;
1070    NORMAL     LDA <251
1080               STA <$7A
1090               LDA >251
1100               STA >$7A
1110    ;
1120    NORMAL1    JMP $A7ED
1130    ;
1140    ;
1150    ;
1160    ;
1170    ;EXECUTE 'COLOR' COMMAND:
1180    ;
1190    ;GET INK PARAMETER.
1200    ;
1210    COLOR.RT   JSR CHARGET
1220               JSR $AD8A
1230               JSR $B7F7
1240    ;
1250    ;CHANGE INK COLOUR.
1260    ;
1270               STY 646
1280    ;
1290    ;GET PAPER PARAMETER.
1300    ;
1310               JSR CHARGET
1320               JSR $AD8A
1330               JSR $B7F7
1340    ;
1350    ;CHANGE PAPER COLOUR.
1360    ;
1370               STY 53281
1380    ;
1390    ;GET BORDER PARAMETER.
1400    ;
1410               JSR CHARGET
1420               JSR $AD8A
1430               JSR $B7F7
1440    ;
1450    ;CHANGE BORDER COLOUR.
1460    ;
1470               STY 53280
1480    ;
1490    ;JUMP TO REST OF ROM-ROUTINE.
1500    ;
1510               RTS
1520    ;
1530    ;
1540    ;
1550    ;TEST FOR REST OF 'OFF'.
1560    ;
1570    OFF.RT     JSR CHARGET
1580               CMP 'F
1590               BEQ OFF.RT1
1600               JMP NORMAL
1610    OFF.RT1    JSR CHARGET
1620               CMP 'F
1630               BEQ OFF.RT2
1640               JMP NORMAL
1650    ;
1660    ;EXECUTE 'OFF' COMMAND.
1670    ;
1680    OFF.RT2    JSR EXTBASOFF
1690    ;
1700    ;GET NEXT CHARACTER AND
1710    ;JUMP TO REST OF ROM-ROUTINE.
1720    ;
1730               JSR CHARGET
1740               RTS
1750    ;
1760    ;
1770    ;
1780    ;TRACE ROUTINE:
1790    ;
1800    ;IF DIRECT MODE, EXIT AT ONCE.
1810    ;
1820    TRACE      LDA $9D
1830               CMP #$80
1840               BNE TRACE1
1850               RTS
1860    ;
1870    ;SAVE CURRENT PRINT POSITION.
1880    ;
1890    TRACE1     SEC
1900               JSR PLOT
1910               STX 50000
1920               STY 50001
1930    ;
1940    ;PLOT TOP LINE PRINT POSITION.
1950    ;
1960               CLC
1970               LDX #0
1980               LDY #0
1990               JSR PLOT
2000    ;
2010    ;CLEAR TOP TWO LINES.
2020    ;
2030               LDA #<EMPTYLINE
2040               LDY #>EMPTYLINE
2050               JSR PRINTSTR
2060    ;
2070    ;PLOT TOP LINE PRINT POSITION.
2080    ;
2090               CLC
2100               LDX #0
2110               LDY #0
2120               JSR PLOT
2130    ;
2140    ;PRINT LINÉ NUMBER.
2150    ;
2160               LDA $3A
2170               LDX $39
2180               JSR PRINTNO
2190    ;
2200    ;PRINT ONE SPACE.
2210    ;
2220               LDA #32
2230               JSR PRINT
2240    ;
2250    ;GET ADDRESS OF BASIC TOKEN AND
2260    ;PRINT IT.
2270    ;
2280               LDY #0
2290               LDA ($7A),Y
2300    ;
2310    ;IF TOKEN, PRINT IT & JUMP FORW.
2320    ;
2330               CMP #128
2340               BCC NOTOKEN
2350    ;
2360               JSR TOKENSR
2370    ;
2380               JMP TOKENDONE
2390    ;
2400    ;IF NO TOKEN, PUT CHARGET ADDRESS
2410    ;INTO 253/254, AND DON'T INCREM.
2420    ;
2430    NOTOKEN    LDA <$7A
2440               STA <253
2450               LDA >$7A
2460               STA >253
2470    ;
2480               JMP TRACE2
2490    ;
2500    ;PUT CHARGET ADDRESS INTO 253/254
2510    ;AND INCREMENT BY ONE.
2520    ;
2530    TOKENDONE  LDA <$7A
2540               STA <253
2550               LDA >$7A
2560               STA >253
2570    ;
2580               INC <253
2590               BNE TRACE2
2600               INC >253
2610    ;
2620    ;PRINT REST OF LINE.
2630    ;
2640    TRACE2     LDY #0
2650    PRINTLOOP  LDA (253),Y
2660               BEQ LINEEND
2670               CMP ':
2680               BEQ LINEEND
2690               CMP #128
2700               BCC PRINTLOO1
2710    ;
2720    ;IF TOKEN, SAVE Y AND PRINT TOKEN
2730    ;
2740               INY
```

```
2750            STY $63
2760  ;
2770            JSR TOKENSR
2780  ;
2790  ;POINT AT CHARACTER AFTER TOKEN
2800  ;AND LOOP BACK.
2810  ;
2820            CLC
2830            LDA $63
2840            ADC <253
2850            STA <253
2860            BCC TRACE2
2870            INC >253
2880            BNE TRACE2
2890  ;
2900  ;PRINT ORDINARY CHARACTER, INCR.
2910  ;INDEX AND LOOP BACK.
2920  ;
2930  PRINTLOO1 JSR PRINT
2940            INY
2950            BNE PRINTLOOP
2960  ;
2970  ;PRINT CARRIAGE RETURN.
2980  ;
2990  LINEEND   JSR PRINTCR
3000  ;
3010  ;WAIT FOR KEYPRESS.
3020  ;
3030  WAIT      JSR $FFE4
3040            BEQ WAIT
3050  ;
3060  ;IF 'V' PRESSED, PRINT VARIABLES.
3070  ;
3080            CMP 'V
3090            BNE ALLDONE
3100  ;
3110            JSR VARS.RT
3120  ;
3130  ;WAIT SOME MORE.
3140  ;
3150            JMP WAIT
3160  ;
3170  ;RECOVER OLD PRINT POSITION
3180  ;
3190  ALLDONE   CLC
3200            LDX 50000
3210            LDY 50001
3220            JSR PLOT
3230  ;
3240  ;IF KEY PRESSED, EXIT.
3250  ;
3260            RTS
3270  ;
3280  ;
3290  ;
3300  ;TOKEN SUBROUTINE:
3310  ;
3320  ;CALCULATE TOKEN NUMBER AND
3330  ;STORE IT IN X.
3340  ;
3350  TOKENSR   SEC
3360            SBC #128
3370            TAX
3380  ;
3390  ;INITIATE BEG. OF TOKEN TABLE.
3400  ;
3410            LDA #<$A09E
3420            STA <$61
3430            LDA #>$A09E
3440            STA >$61
3450  ;
3460  ;IF X=0, TOKEN FOUND.
3470  ;
3480  TOKENLOOP CPX #0
3490            BEQ PRINTOKEN
3500            DEX
3510  ;
3520  ;FIND END OF TOKEN.
3530  ;
3540            LDY #0
3550  TOKENLP1  LDA ($61),Y
3560            CMP #128
3570            BCS ENDTOKEN
3580            INY
3590            BNE TOKENLP1
3600  ;
3610  ;POINT TO BEG. OF NEXT TOKEN.
3620  ;
3630  ENDTOKEN  CLC
3640            INY
3650            TYA
3660            ADC <$61
3670            STA <$61
3680            BCC TOKENLOOP
3690            INC >$61
3700            JMP TOKENLOOP
3710  ;
3720  ;PRINT BASIC WORD.
3730  ;
3740  PRINTOKEN LDY #0
3750  PRINTOK1  LDA ($61),Y
3760            CMP #128
3770            BCS TOKENEND
3780            JSR PRINT
3790            INY
3800            BNE PRINTOK1
3810  ;
3820  ;PRINT LAST CHAR. OF BASIC WORD,
3830  ;AFTER HAVING SUBTRACTED 128.
3840  ;
3850  TOKENEND  SEC
3860            SBC #128
3870            JSR PRINT
3880  ;
3890            RTS
3900  ;
3910  ;
3920  ;
3930  ;PRINT VARIABLES ROUTINE:
3940  ;
3950  ;PUT START OF VARIABLES AREA
3960  ;INTO 251/252.
3970  ;
3980  VARS.RT   LDA <$2D
3990            STA <251
4000            LDA >$2D
4010            STA >251
4020  ;
4030  ;THIS IS THE MAIN LOOP:
4040  ;
4050  ;IF END OF VARS AREA, EXIT.
4060  ;
4070  VARSLOOP  LDA <251
4080            CMP <$2F
4090            BNE VARSLOOP1
4100            LDA >251
4110            CMP >$2F
4120            BEQ VARSEXIT
4130  ;
4140  ;GET NAME OF VARIABLE, TEST IF IT
4150  ;IS FLOATING POINT, INTEGER OR
4160  ;STRING VARIABLE, AND PRINT IT.
4170  ;
4180  VARSLOOP1 LDY #0
4190            LDA (251),Y
4200            CMP #128
4210            BCS INTEGER
4220            JSR PRINT
4230  ;
4240            INY
4250            LDA (251),Y
4260            CMP #128
4270            BCS STRING
4280            CMP #0
4290            BEQ FLPOINT
4300            JSR PRINT
4310            JMP FLPOINT1
4320  ;
4330  ;PRINT FLOATING POINT VARIABLE.
4340  ;
4350  ;PRINT ' = '.
4360  ;
4370  FLPOINT   LDA #32
4380            JSR PRINT
4390  ;
4400  FLPOINT1  LDA #<TEXT1
4410            LDY #>TEXT1
4420            JSR PRINTSTR
4430  ;
4440  ;PUT 251/252 INTO A/Y AND INCRM.
4450  ;BY 2 TO POINT AT CONT. OF VARS.
4460  ;
4470            LDA <251
4480            LDY >251
4490  ;
4500            CLC
4510            ADC #2
4520            BNE FLPOINT2
4530            INY
4540            LDY >251
4550  ;
4560  ;PUT CONTENTS OF VARIABLE INTO
4570  ;FLOATING POINT ACCU 1.
4580  ;
4590  FLPOINT2  JSR $BBA2
4600  ;
4610  ;CONVERT FLOATING POINT ACCU INTO
4620  ;ASCII STRING.
4630  ;
4640            JSR $BDDD
4650  ;
4660  ;PRINT THAT STRING.
4670  ;
4680            JSR PRINTSTR
4690  ;
4700  ;
4710  ;CONTINUATION OF MAIN LOOP:
4720  ;
4730  ;GET NEXT VARIABLE.
4740  ;
4750  VARSLOOP2 JSR PRINTCR
4760  ;
4770            CLC
4780            LDA <251
4790            ADC #7
4800            STA <251
4810            LDA >251
4820            ADC #0
4830            STA >251
4840  ;
4850            JMP VARSLOOP
4860  ;
4870  ;
4880  ;ALL DONE: EXIT
4890  ;
4900  VARSEXIT  RTS
4910  ;
4920  ;
4930  ;
4940  ;PRINT INTEGER VARIABLE:
4950  ;
4960  ;PRINT NAME OF INTEGER VARIABLE.
4970  ;
4980  INTEGER   SEC
4990            SBC #128
5000            JSR PRINT
5010  ;
5020            INY
5030            LDA (251),Y
5040            SEC
5050            SBC #128
5060            JSR PRINT
5070  ;
5080  ;PRINT '% = '.
5090  ;
5100            LDA #<TEXT2
5110            LDY #>TEXT2
5120            JSR PRINTSTR
5130  ;
5140  ;PRINT CONTENTS OF VARIABLE.
5150  ;
5160            LDY #2
5170            LDA (251),Y
5180            PHA
5190            INY
5200            LDA (251),Y
5210            TAX
5220            PLA
5230            JSR PRINTNO
5240  ;
5250  ;ALL DONE: BACK TO MAIN LOOP.
5260  ;
5270            JMP VARSLOOP2
5280  ;
5290  ;
5300  ;
5310  ;PRINT STRING VARIABLE:
5320  ;
5330  ;PRINT '$ = '.
5340  ;
5350  STRING    SEC
5360            SBC #128
5370            JSR PRINT
5380  ;
5390            LDA #<TEXT
5400            LDY #>TEXT
5410            JSR PRINTSTR
5420  ;
5430  ;STORE LENGTH OF STRING IN 255.
5440  ;
5450            LDY #2
5460            LDA (251),Y
5470            STA 255
5480  ;
5490  ;POINT 253/254 AT BASE OF STRING.
5500  ;
5510            INY
5520            LDA (251),Y
5530            STA <253
5540            INY
5550            LDA (251),Y
5560            STA >253
5570  ;
5580  ;PRINT STRING.
5590  ;
5600            LDY #0
5610  PRINTLOO2 LDA (253),Y
5620            JSR PRINT
5630            INY
5640            CPY 255
5650            BNE PRINTLOO2
5660  ;
5670  ;ALL DONE: BACK TO MAIN LOOP.
5680  ;
5690            JMP VARSLOOP2
5700  ;
5710  ;
5720  ;
5730  TEXT      BYT "$ = ",0
5740  TEXT1     BYT "  =",0
5750  TEXT2     BYT "% = ",0
5760  EMPTYLINE BYT "                         "
5770            BYT "                         "
5780            BYT "                         "
5790            BYT "                         "
5800            BYT "                   ",0
```

YOUR COMMODORE

Lineage: 53p per word.
(+ VAT)
Semi display: £11.50 plus VAT per single column centimetre minimum 2cm. Ring for information on series bookings/discounts.

All advertisements in this section must be prepaid.
Advertisements are accepted subject to the terms and conditions printed on the advertisement rate card (available on request).

0442 66651

Send your requirements to:
CLASSIFIED DEPARTMENT
ASP LTD. ARGUS HOUSE,
BOUNDARY WAY,
HEMEL HEMPSTEAD HP2 7ST.

SOFTWARE

ONLY POOLS AND HORSES

FOOTBALL BOXFORM £14.95 (+ **£2 for 3"** disc)

★ The **ONLY** pools program written by a **GENUINE EXPERT** who has been professionally employed as such by the leading pools company.
★ The **ONLY** program dealing with football form that has its team merit ratings published in a national sporting magazine.
★ The **ONLY** program that uses machine code and advanced statistical methods to analyse football form. Lists homes aways and draws in order of merit and gives true odds for every match. Covers league & non-league clubs
★ The **ONLY** programm to claim **50% MORE DRAWS** than obtained by chance.

BOXFORM & HANDICAP WINNER £14.95 (+ £2 for 3" disc)

Two programs fro the price of one. Easy to use and they never go out of date. All you need is a few minutes with a daily paper. **BOXFORM** weighs up a race by considering the many facets of form. Easily amended to suit the user's ideas. **HANDICAP WINNER** shows the value for money bets. **OVER 500 WINNERS** in handicaps this year at prices up to 25/1.
SPECIAL PRICE of £24.95 (+ £2 for 3" disc) for all of the above programs

RACING TOOL KIT £49.95 tape or disc

A package of 7 programs for the racing enthusiast. BOXFORM & HANDICAP WINNER plus programs to produce ratings and time figures, update private handicaps or form your own, keep records, work out almost any bet etc. etc.

Why pay inflated prices? **BOX**oft **CLEVER** and get the **BEST**

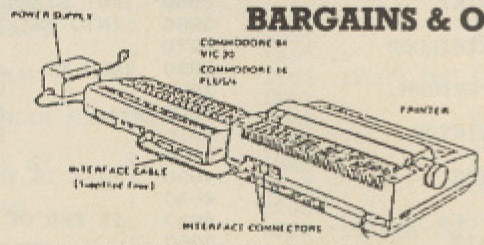
Available on tape or disc for SPECTRUM, CMD C64/128, AMS CPC's, BBC B. from:

BOXOFT (YC), 65 Allans Meadow, Neston, South Wirral. L64 9SQ

Credit card holders may phone 051-336-2668 (24 hrs)

HARDWARE

EEC LTD. CBM PRINTER BARGAINS & OCP ART STUDIO



★SEIKOSHA DOT MATRIX £125.00★
GP 500 VC c.p.s. Pin feed with CBM Serial connector for Commodores (as shown above for HRIC)

★CBM, OCP★
ART STUDIO £12.95 ADVANCED VERSION £19.95
a further £2 off if ordered with printer

★SPECIAL OFFER COMMODORE 64 P.S.U. £22 inc post ★

All prices include VAT Add £6 postage for printers. Other items £2. Tel orders VISA & ACCESS or cash with order. Ansaphone orders ANYTIME Please leave telephone number for acknowledgement.

★ BROTHER HR10C★
★ £129.95 ★
daisy wheel 60 col
12 CPS 2K Buffer,
FEED • BOOKS IN ENGLISH, GERMAN & FRENCH
IDEAL FOR WORD PROCESSING
Previously list price
£343.00

18-21 Misbourne House,
Chiltern Hill,
Chalfont St Peter, Bucks.
SL9 9UE.
Tel: 0753 888866

YOUR COMMODORE SPECIALS

TECH DRAW 64 - A comprehensive technical illustration aid for C64 owners for both disk and tape. Commands available include LINE DRAW, RAYS, BOX, CIRCLE, ARC, ELLIPSE, FREEHAND, FILL, HATCH & TEXT. Wipe copy, move and rotate, save and load or print your drawing all for £8.95 tape or disk. Order Codes TD64T (tape) TD64D (disk)
Full details in Your Commodore July 1987.

SPEEDY ASSEMBLER
Your Commodore's very own Assembler, a 100% memory resident program for loading from tape or disk, featured in the Your Commodore Machine Code Series and now the Y.C. standard assembler only £9.95 Tape or Disk. Order Codes YSPASSD (disk) YSPASSR (tape)
Order from Your Commodore Readers Services at 9, Hall Road, Maylands Wood Estate, Hemel, Hempstead Herts HP2 7BH. Please make cheques payable to A.S.P. Ltd., or telephone your Access/Visa order on 0442 41221

COMPUTER GAMES

To gamble free of charge try the Grand National. Win £5000 or go to Bankrupt. On tape for Commodore +4.
For further details ring
0292 611566

PUBLIC DOMAIN SOFTWARE for the C64/128.

We have 150 disks full of all types of Programs
All disks are £2.95 each
NOW FORTH +4
from £3.75 per disk.
Send SAE or phone for free catalogue (state model)
Kingsway Computer Services
Dept 15 140 Rushdale Road, Sheffield S8 9QE Tel: (0742) 588429

SAMDALE COMPUTERS LTD

"Better than the best of the rest"

RING NOW Credit Card Hotline 0621 742617

THE LATEST SOFTWARE FOR COMMODORE C64/C128/AMIGA

Ring now for a free price list on 0621 742617.
or send stating name and address and machine type.
SAMDALE COMPUTERS LTD
1a Austral Way, Highfield Rise, Althorne, Essex CM3 6DN.

SERVICES

COMMODORE SUPPLIES

C64 Power Supply	£19.95
CBM Plus 4 Power Supply	£29.95
C-16 Power Supply	£12.95
1530 Datasette	£27.95
1531 datasette	£27.95
Simons Basic Extension	£9.95
Serial Lead (for d/drive/printer)	£5.95
C64 reset Switch	£4.95
Dust Cover (please state computer)	£3.50

Prices include VAT & P&P. Allow up to 14 days for delivery. Chq/PO to:
Omnidale Ltd (Y.C.) 23 Curzon Street, Derby, DE1 2ES. Tel 0332 291219.

ADVERTISERS

TO ADVERTISE IN YOUR COMMODORE CALL 0442 66551

FOR DETAILS ON ADVERTISEMENT RATES CALL 0442 66551

SPECIAL OFFERS

64/128 USERS.

SPECIAL OFFER. **ONLY £40 VAT inc. MODEM/DATACOMS CARD,**
ASCII terminal facilities. Automatic dialling autoanswering, spooling and printing files. In Viewdata mode terminal becomes host computer downloading text with local graphics editing. Answer mode baud rate recognition. Telex/Mail dumping to disk.
Cheques to:
AEL, Barclays Bank Chambers, Keswick CA12 5BJ Visa or Access
Tel. 07687 74439 *Post/Pack £1.00*

"You better believe OUR PRICES"

Amiga 500 + 10 Star games + gifts	£389.00
Commodore 1084S Colour Monitor	£240.00
C64 Compatible Data Recorder	£24.50
Commodore 1351 C64/C128 Mouse	£19.99
C64 Compatible Power Supply	£19.99
C64 Slimline Replacement Case	£20.00
Commodore 1541 C64 $5\frac{1}{4}$" D/Drive	£159.95
Oceanic $5\frac{1}{4}$" D/Drive + FREE S/Ware	£119.95
3.5" Disks x 10	£9.50
5.25" Disks x 10	£4.50
Disk Box 80 x 3.5" (Lockable)	£6.99
Disk Box 100 x 5.25" (Lockable)	£6.99

ALL PRICES INCLUDE VAT
P/P. £1.50 SMALL ITEMS, £3.50 FOR HARDWARE

Crofton Micro Supplies
FREEPOST, London. SE4 2BR.
Telephone 01-469-3246

COMMODORE REPAIR SERVICES

TO ADVERTISE YOUR COMMODORE REPAIR SERVICES CALL: JAYNE HEIGOLD ON 0442 66551

RATE: £11.50 (+ VAT) PER SINGLE COLUMN CM
STANDARD SIZE 4cm x 1 col = £46.00 (+VAT)

TELEPHONE BOOKINGS WELCOME

VTS COMPUTER REPAIRS

01 - 407 9829

Simply send your unit to the address below securely packed with fault description and max fee.
(COMMODORE, AMSTRAD, BBC, SINCLAIR + MORE.........**£28.00**

VTS COMPUTER REPAIRS

12 Lyly House, Gt. Dover Street, London, SE1 4EQ.
FAST, RELIABLE TURNAROUND

M.C.E. SERVICES

33, Albert Street, Mansfield, Notts
NG18 1EA Tel: 0623 653512

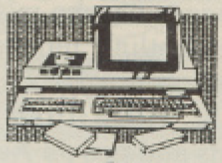
FIXED PRICE COMMODORE REPAIRS

C64	£24.95
C16	£19.95
PLUS4	£24.95
AMIGA A.500	£29.95
1541	£29.95
1571	£31.95
C2N etc	£12.95

All prices inc. VAT, Labour, Parts etc.
NO MORE TO PAY. Please ring for details.

COMPUTER SYSTEMS

★ REPAIRS ★ SPARES
★ SALES & SUPPLIES



dB
ELECTRONIC SERVICES
CHELMSFORD 260874

COMMODORE SPARES & REPAIRS

REPAIRS

C64, C+4 .... £25 inc.
C128, 1541 .... £32 inc.

Three month warranty, 1 week turnround

SPARES

906114	£9.30	901227	£12.95
6510	£9.95	8501	£8.95
6526	£9.50	C64 Power Pack	£22.00
6581	£11.25	C64 Cassette Unit	£22.00
6569	£21.50	C128 Power Pack	£69.95

Inc Vat & PP

OASIS COMPUTER SERVICES
Dept 4C, 14 Ridgeway Road, Salisbury, Wiltshire, SP1 3BU.
Tel (0722) 335061

HINDLEY ELECTRONICS

SAME DAY SERVICE (CALLERS) C64 ONLY

FAST

24hr TURNAROUND (POSTAL) C64 ONLY

EXPERT COMPUTER REPAIRS
COMMODORE - SINCLAIR - ACORN
STANDARD CHARGES
(INC VAT AND FULLY INSURED RETURN POSTAGE)

Commodore 64	£25.00	Commodore 64C	£29.00	Commodore 128	£32.00
Commodore +4	£25.00	Commodore vic 20	£18.00	AMIGA 500	£40.00
1541 disc drive	£32.50	(OTHER DRIVES REPAIRED RING FOR DETAILS			
Spectrum 48k	£18.00	Spectrum + 2	£25.00	Acorn Electron	£21.00

AMIGA 500 Disk Drive Replacement **£65.00**

3 MONTHS "NO QUIBBLE" WARRANTY ON ALL REPAIRS
SPARES — SPARES — SPARES — SPARES
Low prices E.G. Spectrum 48k MEMBRANE £3.99 Spectrum £6.99 LOW PRICES
COMPUTER ICS and GENERAL COMPONENTS WHY PAY MORE (Please ring before posting goods)

HINDLEY ELECTRONICS
DEPT YC, 97 MARKET STREET, HINDLEY WIGAN WN2 3AA Tel (0942) 522743

Computer repairs: Callers requiring same day service please ring before calling and add 10% to standard charge - Standard charges do not include repairs to external items E.G power units tape recorders joysticks etc. A surcharge may be levied for machines that have sustained severe damage E.G fire flood tamper

TERMS & CONDITIONS

CLASSIFIED ADVERTISING TERMS & CONDITIONS

Our terms for new advertisers (semi-display and lineage) are strictly pro-forma payments until satisfactory reference can be taken up (excluding recognised advertising agencies). Cheques and P.O.'s should be crossed and made payable to ARGUS SPECIALIST PUBLICATIONS and sent to:
THE CLASSIFIED DEPT
ARGUS HOUSE, BOUNDARY WAY, HEMEL HEMPSTEAD HP2 7ST.
There are no reimbursements for cancellations. Advertisements arriving too late for a particular issue will be inserted in the following issue unless accompanied by instructions to the contrary. It is the responsibility of the advertiser to ensure that the first insertion of every series is published correctly, and corrections must be notified in time for the second insertion, otherwise the publishers will not accept liability or offer any reduction in charges.
All advertising sales are subject to Government Regulations concerning VAT. Advertisers are responsible for complying with the various legal requirements in force eg: The Trade Description Act, Sex Discrimination Act & the Business Advertisements (disclosure) order 1977.
Full Terms & Conditions of Advertising available on request.

FOR MORE DETAILS OF RATES AND SERIES DISCOUNTS CALL JAYNE HEIGOLD ON THE CLASSIFIED *HOTLINE* 0442 66551

PROBLEMS WITH YOUR COMMODORE ?

FOR FAST, RELIABLE AND PROFESSIONAL REPAIRS AT COMPETITIVE PRICES

P M ENGINEERING

UNIT 260, MILTON SCIENCE PARK, CAMBRIDGE CB4 4WE
ST. IVES (0223) 420007

We can also supply Hardware, Software, Blank Disks, Accessories and Spare parts.
C64 Repairs £25.00 inc. VAT (p&p FREE)

The End is Nigh..

...But not with us yet as, like most of the rest of the magazine, BREAK is to leave us as we approach a new era. This is now the page where you'll be able to read all the juicy gossip that is floating around the Commodore market.

The Lines are Engaged...

A Reading resident has been having a rather harassing time of late due to an Amstrad magazine who misprinted the Activision hotline number. The person in question was receiving calls all around the clock from people requesting hints and tips on Activision games – causing some distress. The correct number is (0734) 310003.

Speaking of telephone lines, those kind peeps at British Telecom have changed Incentive's number. The new number and code is, if you're at all interested, (0734) 817288.

Arise Sir Jez

Jez Ford, the editor of Electronics Today International – one of our many sister magazines, flew the Argus flag at the recent Lone Wolf press launch, when he came away with the prize in a novel competition. Out of the thirty plus press representatives Jez was the first to solve many clues and discover the identity of Lone Wolf (a teenage thug in costume).



Of course, if it wasn't for a certain Deputy Editor's blatant cheating, the result may have turned out very different.

Just When You...

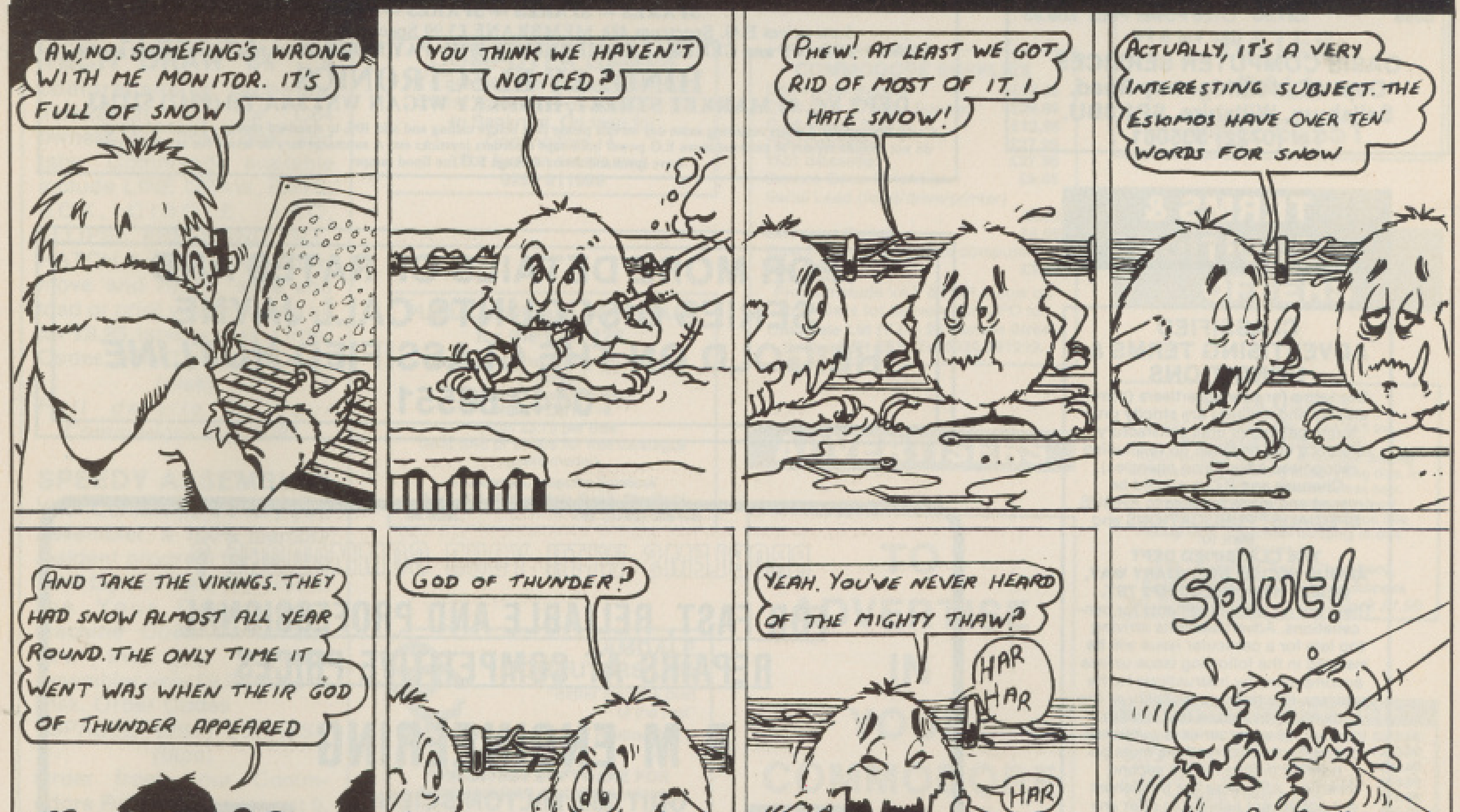
...Thought you could get your hands on Mars Cops, the game has been delayed, yet again.

Due originally for release almost a year ago, the game was nearing completion when the fickle finger of fate struck Arcana Software Limited. The programmer, who had worked to getting the game two weeks from being ready, has walked out on the company. This means a new programmer will have to finish the project and thus we shall not see Mars Cops until the beginning of September.

Commodore Where Are You?

As we constantly get asked for the telephone number of Commodore U.K., it is best if we just print it here: Tel: (0628) 770088.

The Nibbles *By Alan Batchelor*



GREENPEACE



NUCLEAR POWER

The possibility of future nuclear accidents is real.
The probability is high.
A core meltdown accident possibly resulting in Chernobyl scale releases of radiation can be expected within 20 years according to former US Nuclear Regulatory Commissioner Asselstine.

Greenpeace has raised fundamental questions on the safety of UK gas-cooled reactors and is campaigning for their closure.

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE



THE GREAT WHALES

Since commercial whaling was banned in 1986, 11,000 whales have been slaughtered, many for so-called scientific research.

Iceland was the first whaling nation to side-step the commercial whaling ban. To make Iceland pay dearly for the whales she kills, Greenpeace is promoting a boycott of Icelandic fish. Meanwhile Greenpeace will work for a permanent ban on commercial whaling.

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE



NON-VIOLENCE

In 17 years of campaigning, Greenpeace's greatest achievement has been its refusal to resort to violence. Ever.

At least as important as opposing nuclear weapons, saving whales or trying to stop ocean pollution, is the way it has been done.

To Greenpeace, the peace is as important as the green: the means is as important as the end.

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE



THE IRISH SEA

The Irish Sea is the world's most radioactively contaminated sea as well as being one of the most poisoned – by sewage and industrial wastes.

Many of its beaches are polluted, fish disease is widespread, seals are dying, dolphins and sea birds have fled their destroyed habitats.

Now Greenpeace has launched a campaign to save the Irish Sea, demanding an end to radioactive discharges and to sea dumping of toxic wastes and sludges.

Greenpeace has sent its ship – the Moby Dick – to confront the polluters and press its demands for a clean Irish Sea.

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE



LOST DOLPHINS

Britain has lost many of her coastal dolphins and porpoises. Pollution, over-fishing and incidental catches are most likely to blame.

To stem the decline and protect existing stocks, Greenpeace has set up its Dolphin Look-Out network and is working to establish protection zones in areas where dolphins and porpoises are still to be found.

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE



ACID RAIN

Acid rain causes death and damage to forests, lakes, wildlife, crops, buildings and ourselves.

Now scientists have calculated the reductions in acid-forming pollutants that must be made.

They are: 90% cuts in sulphur dioxide
75% cuts in nitrogen oxides
75% cuts in atmospheric ozone.

These are the cuts nature demands if it is to survive.
They are the only cuts which Greenpeace will accept.

THANK GOD SOMEONE'S MAKING WAVES

THANK GOD SOMEONE'S MAKING WAVES

GREENPEACE

GREENPEACE URGENTLY NEEDS YOUR SUPPORT.

Please send a personal donation of £12 or a family donation of £17.50. Or more. In return you will receive our quarterly newsletter and campaign updates. Remember, the more you can afford to send us, the more we can do to protect the natural world.

Please accept my donation of: ☐ £12 Single ☐ £17.50 Family ☐ Additional donation £____

☐ I enclose cheque/P.O. for £________ payable to Greenpeace Ltd.

☐ Please charge my Visa/Access a/c no: ☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐☐

Signature: ____________ Date: ____________

Name: Mr/Mrs/Ms ____________

Address: ____________

BC 012

Postcode: ____________

Please return to: Greenpeace, FREEPOST, 30-31 Islington Green, London N1 8BR.